



Universidad
Carlos III de Madrid

Grado de Ingeniería Informática

TRABAJO FIN DE GRADO

Integración de un módulo de gestión de métrica Redmine dentro de un sistema de información corporativo.

Autor: Víctor Manuel Valle Sanz

Tutor: Israel González Carrasco

Resumen

El trabajo que se presenta en este documento es el resultado del Trabajo de Fin de Grado en empresa realizado en colaboración con la empresa SATEC (Sistemas Avanzados de Tecnología, S.A.) y la Universidad Carlos III de Madrid.

El presente documento supone el plan del proyecto “Integración de un módulo de gestión de métrica Redmine dentro de un sistema de información corporativo”. Dicho proyecto se propuso a nivel interno de la empresa para mejorar un proceso. El componente a desarrollar, ofrecerá a los usuarios de la herramienta Redmine la posibilidad gestionar métricas sobre la información que se aloja en este sistema de información, destinado a la gestión de proyectos software.

A lo largo del documento, se detallará el funcionamiento del sistema de información sobre el que se aplicará el componente, así como las tareas del ciclo de vida del proyecto que se va a llevar a cabo.

Además, se realizará un análisis breve del lenguaje de programación Ruby y del *framework* Ruby on Rails con el cual está creada dicha herramienta.

Por último, se adjunta como anexo un resumen en inglés que contiene la introducción, con los objetivos del proyecto y las conclusiones del mismo.

Summary

The work presented in this document is the result of the Final Grade, which was done in collaboration with the company SATEC (Sistemas Avanzados de Tecnología, S.A.) and the Carlos III University of Madrid.

This document presents the project plan "Integration of a management module Redmine metric within a corporate information system. This project aimed to improve a process at the internal level of the company. The component to be developed, will offer users the possibility to manage tool Redmine metrics on the information that is hosted on this information system aimed project management software.

Throughout the document, operation of the system on which the component is applied will be explained and the tasks of the project life cycle to be performed.

In addition, a short analysis of the Ruby programming language and Ruby on Rails framework with which Redmine was created is included.

Finally, a summary in English containing the introduction, with the objectives of the project and its conclusions are annexed.

Índice

1 Introducción.....	12
1.1 Redmine	13
1.2 Lenguaje de programación Ruby	14
1.3 Framework Ruby on Rails	15
1.4 Objetivos del proyecto	17
1.5 Metodología del proyecto	18
1.6 Estructura del documento	19
2 Estudio de viabilidad	21
2.1 Introducción	21
2.2 Descripción del Sistema actual	21
2.3 Definición del Sistema a desarrollar	23
2.3.1 Alcance	23
2.3.2 Identificación de los usuarios participantes	25
2.4 Estudio de la situación actual	25
2.4.1 Assembla	26
2.4.2 Redbooth	26
2.5 Estudio de alternativas de solución	27
2.5.1 Representación de gráficos	27
2.5.1.1 Morris.js	27
2.5.1.2 Google Charts	27
2.5.1.3 Highcharts	27
2.5.1.4 Chartkick	28
2.5.1.5 Gruff	28
2.5.2 Generación de documentos PDF	28
2.5.2.1 Prawn	28
2.5.2.2 PDFKit	28
2.5.2.3 Wicked PDF	28
2.5.3 Selección de la solución	29
2.5.3.1 Representación de gráficos	29
2.5.3.2 Generación de documentos PDF	29
3 Gestión de proyecto.....	30

3.1	Planificación	30
3.1.1	Estudio de viabilidad	31
3.1.2	Análisis	32
3.1.3	Diseño	33
3.1.4	Desarrollo	34
3.1.5	Pruebas	35
3.1.6	Resto del proyecto	36
3.1.7	Hitos	37
3.2	Costes	37
3.2.1	Equipo informático	38
3.2.2	Licencias Software	38
3.2.3	Material fungible	38
3.2.4	Coste total del proyecto	38
4	Análisis.....	40
4.1	Introducción	40
4.1.1	Objetivos y restricciones	40
4.2	Catálogo de Requisitos	41
4.2.1	Requisitos de usuario	41
4.2.2	Requisitos Funcionales	41
4.2.3	Requisitos No Funcionales	45
4.3	Subsistemas y Casos de Uso	46
4.3.1	Modelo de Subsistemas	46
4.3.2	Modelo de Casos de Uso	48
4.4	Modelo Lógico de Datos	55
4.5	Interfaz de Usuario	56
4.5.1	Menús	56
4.5.1.1	Menú superior	56
4.5.1.2	Menú de proyecto	57
4.5.1.3	Configuración de métricas	57
4.5.1.4	Añadir métrica	58
4.5.1.5	Editar métrica	61
4.5.2	Visualización de métricas	62

4.5.3	Configuración de permisos	64
4.5.4	Matrices de trazabilidad	66
4.5.4.1	Requisitos	66
4.5.4.2	Requisitos – Casos de uso	71
4.5.4.3	Requisitos – Módulos	72
4.5.4.4	Requisitos – Pruebas	74
4.5.4.4.1	Pruebas 1 - 10	74
4.5.4.4.2	Pruebas 11 - 20	75
4.5.4.4.3	Pruebas 21 - 30	77
4.5.4.4.4	Pruebas 31 - 40	78
4.5.4.4.5	Pruebas 41 - 50	80
4.5.4.4.6	Pruebas 51 - 60	81
4.5.4.4.7	Pruebas 61 - 70	83
4.5.4.4.8	Pruebas 71 - 80	84
4.5.4.4.9	Pruebas 81 - 91	86
5	Diseño.....	88
5.1	Introducción	88
5.2	Definición de la Arquitectura del Sistema	88
5.2.1	Definición de la Arquitectura Software	88
5.2.1.1	Diagrama de subsistemas	90
5.2.1.2	Diagramas de secuencia	93
5.3	Diseño de la Capa de Presentación	96
5.3.1	Diagramas de Navegación	96
5.3.1.1	Subsistema de configuración de métricas	96
5.3.1.2	Subsistema de visualización de métricas	101
5.3.2	Diseño de las Interfaces de Usuario	103
5.4	Diseño de la Capa de Negocio	104
5.4.1	Arquitectura general de la solución	104
5.4.2	Diagramas de Clases de Diseño	105
5.4.3	Especificación de clases	107
5.5	Diseño de la Capa de Datos	114
5.5.1	Modelo Físico de Datos	114
5.5.2	Descripción de las Entidades de Datos	114
5.6	Diseño de Seguridad	115

5.6.1	Definición de usuarios, roles y políticas de acceso	115
5.6.2	Consultas SQL y formularios	116
6	 Plan de pruebas.....	117
7	 Manual de usuario.....	133
7.1	Introducción	133
7.2	Permisos del plug-in	133
7.2.1	Asignación de permisos	133
7.2.2	Activación del módulo de métricas	134
7.2.3	Script de activación del módulo	135
7.2.4	Administrador de métricas	136
7.3	Configuración de métricas	137
7.3.1	Crear nueva métrica	138
7.3.1.1	Tipos de gráficos	141
7.3.2	Editar métrica	143
7.3.3	Eliminar métrica	144
7.4	Visualización de métricas	145
7.4.1	Métricas generales	145
7.4.2	Métricas de proyecto	147
7.5	Exportaciones	147
7.5.1	CSV	148
7.5.2	PDF	148
8	 Manual de despliegue.....	150
8.1	Introducción	150
8.1.1	Consideraciones previas a la instalación	150
8.1.2	Directorios de la aplicación	150
8.2	Usuarios	150
8.3	Instalación de gemas	151
8.3.1	Agregación de gemas	151
8.3.2	Instalación de gemas	151
8.4	Instalación del plug-in	152
8.4.1	Archivos del plug-in	152

8.4.1.1	Directorio métricas	152
8.4.1.2	Directorio pdfs	152
8.4.1.3	Script de activación del módulo	153
8.4.2	Migración de datos	153
8.5	Comprobación de la instalación	153
8.6	Rollback	154
9	 Conclusiones	157
9.1	Resultados de la puesta en producción	157
9.2	Futuras mejoras	158
9.3	Impresiones personales	158
10	 Anexo. Abstract en Inglés del Plan de Proyecto.	160
10.1	Introduction	160
10.1.1	Redmine	161
10.1.2	Ruby language programming	161
10.1.3	Framework Ruby on Rails	163
10.1.4	Project objectives	164
10.1.5	Project Methodology	166
10.1.6	Document Structure	167
10.2	Conclusions	168
10.2.1	Results in production	168
10.2.2	Future improvements	169
10.2.3	Personal impressions	170
11	 Glosario de términos	171
12	 Bibliografía.....	172

Figuras

Ilustración 1: Diagrama del modelo CMMI [10].	18
Ilustración 2: Diagrama del modelo en cascada.	19
Ilustración 3: Logo de Assembla.	26
Ilustración 4: Logo de Redbooth.	27
Ilustración 5: Diagrama de Gantt del proyecto.	30
Ilustración 6: Diagrama de Gantt del estudio de viabilidad.	31
Ilustración 7: Diagrama de la fase de análisis.	32
Ilustración 8: Diagrama de Gantt de la fase de diseño.	33
Ilustración 9: Diagrama de Gantt de la fase de desarrollo.	34
Ilustración 10: Diagrama de Gantt de la fase de pruebas.	35
Ilustración 11: Diagrama de Gantt del resto del proyecto.	36
Ilustración 12: Modelo de subsistemas.	46
Ilustración 13: Diagrama de casos de uso.	48
Ilustración 14: Menú superior de Redmine.	56
Ilustración 15: Menú superior de Redmine modificado.	56
Ilustración 16: Menú de proyecto de Redmine.	57
Ilustración 17: Menú de proyecto de Redmine modificado.	57
Ilustración 18: Interfaz de selección de tipo de métricas.	57
Ilustración 19: Interfaz de configuración de métricas.	58
Ilustración 20: Interfaz de creación de métricas.	58
Ilustración 21: Interfaz de creación de métricas de representación simple.	59
Ilustración 22: Interfaz de creación de métricas de representación multiserie.	60
Ilustración 23: Interfaz de edición de métricas.	61
Ilustración 24: Interfaz de visualización de métricas generales.	62
Ilustración 25: Interfaz de visualización de métricas de proyecto.	63
Ilustración 26: Interfaz de datos de una métrica.	63
Ilustración 27: Interfaz de creación de perfil.	64
Ilustración 28: Interfaz de asignación de permisos.	64
Ilustración 29: Componente de la interfaz de creación de perfil.	65

Ilustración 30: Componente de la interfaz de asignación de permisos.	65
Ilustración 31: Permisos del módulo Métricas.	65
Ilustración 32: Arquitectura MVC.	89
Ilustración 33: Subsistema de configuración de métricas.	90
Ilustración 34: Subsistema de visualización de métricas.	91
Ilustración 35: Subsistema de permisos.	91
Ilustración 36: Diagrama de secuencia de configuración de métricas generales.	93
Ilustración 37: Diagrama de secuencia de configuración de métricas de proyecto.	94
Ilustración 38: Diagrama de secuencia de visualización de métricas generales.	95
Ilustración 39: Diagrama de secuencia de visualización de métricas de proyecto.	95
Ilustración 40: Diagrama de secuencia de permisos.	96
Ilustración 41: Diagrama de navegación de configuración de métricas.	97
Ilustración 42: Diagrama de navegación de creación de métricas.	98
Ilustración 43: Diagrama de navegación de edición de métricas.	99
Ilustración 44: Diagrama de navegación de eliminación de métricas.	100
Ilustración 45: Diagrama de navegación de visualización de métricas generales.	101
Ilustración 46: Diagrama de navegación de visualización de métricas de proyecto.	102
Ilustración 47: Diagrama de clases.	105
Ilustración 48: Modelo físico de datos.	114
Ilustración 49: Asignación de permisos.	134
Ilustración 50: Permisos del módulo métricas.	134
Ilustración 51: Activación de módulos.	135
Ilustración 52: Creación del administrador de métricas.	136
Ilustración 53: Permisos del administrador de métricas.	137
Ilustración 54: Acceso al menú de configuración de métricas	137
Ilustración 55: Menú de configuración de métricas.	138
Ilustración 56: Configuración de métricas generales.	138
Ilustración 57: Formulario de creación de una métrica.	139
Ilustración 58: Representación simple.	140
Ilustración 59: Representación multiserie.	140
Ilustración 60: Atributos del gráfico simple.	140

Ilustración 61: Atributos del gráfico multiserie.	141
Ilustración 62: Ejemplo de gráfico de columnas.	142
Ilustración 63: Ejemplo de gráfico de barras.	142
Ilustración 64: Ejemplo de gráfico de líneas.	142
Ilustración 65: Ejemplo de gráfico de áreas.	143
Ilustración 66: Ejemplo de gráfico de tartas.	143
Ilustración 67: Formulario de edición de métricas.	144
Ilustración 68: Mensaje de confirmación de métrica actualizada.	144
Ilustración 69: Eliminación de métrica.	145
Ilustración 70: Mensaje de confirmación de métrica eliminada.	145
Ilustración 71: Acceso al menú de visualización de métricas generales.	145
Ilustración 72: Menú de visualización de métricas generales.	146
Ilustración 73: Visualización de una métrica.	146
Ilustración 74: Acceso al menú de visualización de métricas de proyecto.	147
Ilustración 75: Formatos de exportación.	147
Ilustración 76: Exportación a CSV.	148
Ilustración 77: Exportación a PDF.	149
Ilustración 78: Instalación de gemas.	152
Ilustración 79: Menú superior.	154
Ilustración 80: CMMI model diagram.	165
Ilustración 81: Waterfall model diagram.	167

1 | Introducción

En la actualidad, los sistemas de información se han convertido en una herramienta indispensable a nivel corporativo. Las empresas del sector de la ingeniería de software necesitan que determinadas informaciones fluyan para coordinar sus acciones operativas, y que esta información sea organizada, para una toma de decisiones eficiente. De esta forma se completan las funciones y procesos de negocios que la empresa les ha encomendado para conseguir ventaja competitiva.

Pero, ¿qué es exactamente un sistema de información? Un sistema de información puede definirse como el conjunto de elementos orientados al tratamiento y administración de datos e información (procesamiento de los datos), organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Los elementos que lo componen son los siguientes:

- Equipo humano que desarrolla las actividades encaminadas al cumplimiento del objetivo.
- Los datos destinados a producir la información deseada.
- Las actividades necesarias para cumplir los objetivos.
- Recursos materiales necesarios tanto físicos como tecnologías de la información.

Hoy en día tenemos a nuestra disposición un buen número de herramientas que nos ayudan a ser mucho más productivos con nuestro trabajo, ya sea porque nos permiten mantener ordenada nuestra documentación o nuestras tareas. En la ingeniería de software, dentro del ámbito de la **gestión de proyectos** también las herramientas nos posibilitan la realización de la planificación del proyecto, el seguimiento del mismo y medir el grado de cumplimiento de las tareas del mismo. Todo esto desde una base de datos única donde se organiza la información de los proyectos de la empresa para poder tratarla y alcanzar eficientemente los objetivos marcados.

Entre las herramientas dedicadas a la gestión de proyectos, se pueden encontrar herramientas gratuitas y herramientas de pago. Estas últimas requieren la adquisición de una licencia temporal y normalmente proporcionan una mayor cantidad de funcionalidades que las herramientas gratuitas.

Las empresas buscan un equilibrio entre lo que ofrece una herramienta de pago y lo que ofrece una gratuita y valoran si es necesario invertir presupuesto para mejorar el proceso de desarrollo de software en un entorno colaborativo. En **Satec** (Sistemas Avanzados de Tecnología, S.A.), optaron por utilizar una herramienta gratuita que incluyese suficientes funcionalidades como para poder abarcar las necesidades básicas de la empresa en cuanto a gestión de proyectos. La herramienta elegida fue Redmine.

1.1 Redmine



Redmine es una herramienta en software libre desarrollada en **Ruby on Rails** que nos permite gestionar proyectos, tareas y, además, gestionar errores de software e, incluso, controlar entregas desde el repositorio de un proyecto de desarrollo software [1]. Para ello, Redmine trabaja en base a “peticiones”, cuyo ciclo de vida es gestionado a través de la propia herramienta. Las peticiones de Redmine se clasifican en: Funcionalidad, tarea, error, cambio, mejora, soporte, garantía, peer review de código y auditoría CM.

Incluye herramientas que permiten gestionar los proyectos de forma más visual, como son un calendario de actividades y los diagramas de Gantt, lo cual hace que sea más sencillo comprobar la línea del tiempo de los proyectos.

Cabe destacar, la inclusión de un visor de repositorio de control de versiones (SVN y Git), el cual recoge información de los últimos cambios producidos en él. Además, incorpora para el intercambio de información entre los distintos integrantes del equipo de desarrollo, herramientas como la wiki y el foro, así como la integración con correo electrónico.

Esta herramienta se ha hecho muy popular en muchas empresas y se utiliza para gestionar la planificación de un proyecto, las tareas de éste, los objetivos y la asignación de las tareas a los miembros de un equipo de trabajo; es decir, los requisitos típicos que le pediríamos a una herramienta de esta clase.

Uno de los aspectos que convierten a Redmine en una herramienta muy interesante, es la posibilidad de instalarle plugins que añadan nuevas funcionalidades. Al ser una herramienta open source, muchos desarrolladores han publicado números plugins en la web oficial de Redmine [2]. Por ejemplo, existe un plug-in llamado *Redmine Time Recorder* que permite grabar el tiempo que vamos dedicando a las tareas y proyectos en los que trabajamos. Otro ejemplo es *Redmine Notification Extension*, una extensión para Google Chrome que se encarga de mostrarnos notificaciones si se produce algún cambio en los “defectos” en los que estemos involucrados, es decir, bugs documentados dentro del proyecto de desarrollo software en el que estamos trabajando.

Existe un listado de empresas que utilizan la herramienta Redmine, entre las cuales destacan empresas de software Open Source como Ruby language y TYPO3 Forge, empresas de investigación como el Laboratorio de Investigación de Informática LIM y el Instituto Superior de Aeronáutica y del Espacio. Además, se incluyen empresas del sector público como el ministerio de educación francés para el proyecto EOLE y el gobierno de Chile para su iniciativa de código Open Source [3].

1.2 Lenguaje de programación Ruby



Ruby es un lenguaje de programación dinámico orientado a objetos, creado por Yukihiro Matsumoto y publicado en el año 1995. Su autor recogió características de sus lenguajes favoritos como Perl y Smalltalk para formar un nuevo lenguaje que incorporara tanto la programación funcional como la programación imperativa. El concepto principal que se quiso transmitir, es que Ruby fuese natural, no simple. Es simple en apariencia, pero complejo por dentro, como el cuerpo humano [4].

Ruby es un proyecto de código abierto, lo cual quiere decir que cualquier desarrollador puede usarlo, modificarlo o distribuirlo. Esto también significa que Ruby se mantiene constantemente en un proceso de mejora debido a la gran cantidad de desarrolladores que lo utilizan en todo el mundo.

Las características más importantes de Ruby son:

- **Todo es un objeto:** En Ruby, todo es un objeto. Se le puede asignar propiedades y acciones a toda información y código. La programación orientada a objetos llama a las propiedades variables de instancia y las acciones son conocidas como métodos. En muchos lenguajes, los números y otros tipos primitivos no son objetos. Ruby permite establecer métodos y variables de instancia a todos sus tipos de datos. Esto facilita el uso de Ruby, porque las reglas que se aplican a los objetos son aplicables a todo Ruby.
- **Flexibilidad:** Ruby es un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente. Las partes esenciales de Ruby pueden ser quitadas o redefinidas a placer. Se puede agregar funcionalidad a partes ya existentes. Ruby intenta no restringir al desarrollador.
- **Bloques:** Los bloques de Ruby son también vistos como una fuente de gran flexibilidad. El desarrollador puede anexas una cláusula a cualquier método, describiendo cómo debe actuar. La cláusula es llamada bloque y se ha convertido en una de las más famosas funcionalidades para los recién llegados a Ruby que vienen de otros lenguajes imperativos como PHP o Visual Basic.
- **Mixin:** Ruby se caracteriza por su herencia simple. Sin embargo, Ruby incorpora el concepto de módulos (llamados categorías en Objective-C), que son colecciones de métodos. Las clases pueden mezclar (mixin) un módulo e incorporar todos sus métodos gratuitamente.

El poder de Ruby, viene en parte, de su biblioteca estándar, que define más de 9.000 métodos que se pueden utilizar en el código. Además, incluye a disposición de los programadores librerías llamadas RubyGems (Gemas) las cuales proporcionan muchas más herramientas. Con bastante frecuencia, una aplicación más exigente puede desarrollarse en cuestión de minutos, sólo mediante la investigación y el uso de una gema existente [5].

Desde su aparición, Ruby no ha parado de crecer, atrayendo a desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo y llenando los aforos de las conferencias relacionadas a Ruby. El índice TIOBE, que mide el crecimiento de los lenguajes de programación, ubica a Ruby en la posición número 13 del ranking mundial. Su continuo crecimiento hace predecir que en los próximos meses Ruby se pueda situar entre los diez lenguajes más utilizados. Gran parte de su crecimiento se atribuye a la popularidad alcanzada por aplicaciones desarrolladas con Ruby, en particular el framework de desarrollo web Ruby on Rails [4].

1.3 Framework Ruby on Rails

Ruby on Rails (RoR o Rails) es un framework de desarrollo de aplicaciones web basado en el lenguaje de programación Ruby. El objetivo de este framework es ofrecer a los desarrolladores la posibilidad de realizar aplicaciones web de forma fácil y sencilla [6]. De esta forma, permite realizar una aplicación utilizando menos código que otros lenguajes de programación y frameworks. Los desarrolladores de Rails destacan que el desarrollo de las aplicaciones se hace mucho más ameno que con otras tecnologías.

El desarrollo de aplicaciones web se realiza prácticamente todo desarrollando con el lenguaje de programación Ruby. Permite definir nuestra base de datos y acceder a ella en Ruby, escribir el código de la lógica de negocio en Ruby e insertar código Ruby embebido en los documentos web [7].

Ruby on Rails utiliza la arquitectura **MVC** (Modelo-Vista-Controlador) la cual es uno de los puntos fuertes del framework, ya que proporciona una base sólida sobre la cual se pueden construir aplicaciones. MVC es un patrón de arquitectura de software que ha sido aceptado con el tiempo como uno de los mejores, ya que permite desarrollar, comprender y mantener las aplicaciones de una forma más sencilla. Simplifica la implementación de la aplicación dividiéndola en capas para separar responsabilidades. El modelo supone la capa de datos, la cual contiene la lógica de negocio de la aplicación, mientras que el controlador responde a eventos e invoca peticiones al modelo. Por último, la vista es la capa de presentación, la cual expone la información contenida en el modelo [8].

Ruby on Rails se basa en tres principios filosóficos:

- **DRY:** Don't Repeat Yourself (DRY) es una filosofía que se puede apreciar durante toda la aplicación del framework. Significa que el desarrollador no debería tener que repetirse en el código, la configuración, o en muchos casos, incluso la documentación, dentro de una sola aplicación web. Por ejemplo, en una aplicación hecha con Ruby on

Rails, se define la estructura de la base de datos en un solo lugar. No hace falta archivos SQL que definan la estructura, ya que todo se hace en la parte del modelo de la aplicación en código Ruby. Esto supone un ahorro de tiempo y errores, ya que si existe la necesidad de realizar un cambio en la aplicación solo se tendrá que hacer en un lugar. Se evitará de esta forma tener que buscar y realizar cambios en varios archivos para hacer un cambio en la aplicación.

- **Convenio de configuración:** Rails utiliza un convenio para establecer la configuración de la aplicación. Una de las características de muchos frameworks de aplicaciones web es que requieren una configuración específica a través de ficheros XML. Rails aplica una configuración por convenio la cual se piensa que se debería realizar en la gran mayoría de los casos, permitiendo ser modificada en el caso de que el desarrollador lo crea necesario. Un ejemplo de esta configuración se puede encontrar en el sistema de enrutamiento, a través del cual Rails establece un formato estándar de las URL en base a las clases y métodos que tenga la aplicación.
- **Software dogmático:** Rails incorpora opiniones de cómo deben ser desarrolladas y diseñadas las aplicaciones web. Muchos desarrolladores pueden desear utilizar frameworks libres de opiniones y más flexibles, que les permitan incorporar sus decisiones de diseño. Los desarrolladores de Rails rechazan esta idea y han replanteado una visión sobre cómo se deben desarrollar las aplicaciones.

En la actualidad, muchas aplicaciones web se crean utilizando el framework Ruby on Rails en detrimento de J2EE. Los principales motivos son:

- Requiere **menos tiempo y esfuerzo** para crear una aplicación web y ponerla en funcionamiento.
- Las aplicaciones que se desarrollan con Rails requieren **menos configuración**, si se desea aplicar la configuración que el framework incorpora por defecto.
- Rails requiere **menos código** para realizar las mismas tareas que J2EE gracias a Ruby.
- **Active Record:** Es una herramienta ORM (Object Relational Mapping) encargada de gestionar el modelo en la arquitectura MVC, el cual supone la capa responsable de los datos y la lógica de negocio. Esta herramienta facilita la creación de los objetos del modelo cuyos datos son persistidos en la base de datos. Permite implementar de forma fácil un CRUD de operaciones (create, read, update and delete) de los datos utilizando los métodos de Active Record. Todo esto lo hace mucho más atractivo que las librerías de J2EE como Hibernate las cuales son más potentes y flexibles, pero requieren un mayor conocimiento sobre cómo utilizar el código y cómo mantenerlo.

Los creadores de **Twitter**, la red social basada en microblogging que cuenta, según los últimos datos de Octubre de 2014, con 284 millones de usuarios, eligieron el framework de Ruby on Rails para el desarrollo de su aplicación web. Entre otras aplicaciones que utilizan Ruby on Rails destacan GitHub, Assembla, Shopify, Groupon y Bloomberg.

1.4 Objetivos del proyecto

Cuando llegué al departamento de desarrollo de software de **Satec** (Sistemas Avanzados de Tecnología, S.A.) en Julio de 2014, me encomendaron la labor de realizar un proyecto que les había encargado el departamento de Aplicaciones y Servicios. Querían que crease un **plug-in** para **Redmine**, el cual pudiese ofrecer a los usuarios de la herramienta la posibilidad de obtener datos de los proyectos de forma visual, para una mejor captación de la información. El proyecto se llamó “**Métricas Redmine**”.

En el campo de la ingeniería del software, una métrica es cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software o un sistema de información, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo. En el caso de Redmine, las métricas se realizan en base a consulta de un dato (o conjunto de datos) en concreto sobre un proyecto, el cual es interesante poder representarlo de forma gráfica para ver su evolución. Para ello el plug-in debería ser capaz de ejecutar consultas en lenguaje SQL sobre las base de datos de Redmine (la cual contiene todos los datos de los proyectos que se realizan en Satec), y representar los resultados en forma gráfica y tabular. De esta forma, se podría generar un conjunto de métricas que aportasen diversa información de los proyectos y permitiese conocer su estado de forma visual más rápidamente.

Antes de la propuesta de proyecto, el departamento de Aplicaciones y Servicios establecía estas métricas en una hoja de cálculo, en la cual anotaban los resultados obtenidos de la ejecución de consultas sobre la base de datos de Redmine, la cual estaba montada bajo el sistema gestor de bases de datos MySQL. Una vez tenían los datos, los organizaban en una tabla y los representaban gráficamente. A todo este proceso se le añadía la labor de agregar la tabla y la gráfica a un informe de resultados de los proyectos en formato PDF, para poder entregarlo tanto al director del departamento de desarrollo, como a la gerencia de la empresa que requiriera dicha información.

La propuesta del proyecto resultaba interesante para el departamento de desarrollo, ya que permitía que los jefes de proyecto tuvieran acceso a un resumen de su proyecto a través de los resultados que se pretendían obtener de las métricas. Y además, se buscaba la posibilidad de generar informes de resultados de las métricas en formato PDF directamente desde la aplicación para evitar el tedioso proceso citado anteriormente.

La utilización de este componente en una empresa puede permitir que se alcance el nivel 4 de **CMMI**, ya que se realizarán mediciones que permitirán mejorar los procesos. CMMI (Integración de modelos de madurez de capacidades) es un modelo destinado al logro de operaciones de alto rendimiento. Proporciona un conjunto de prácticas para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software [9]. A continuación se muestra un diagrama que indica los cinco niveles del modelo que puede adquirir una organización:

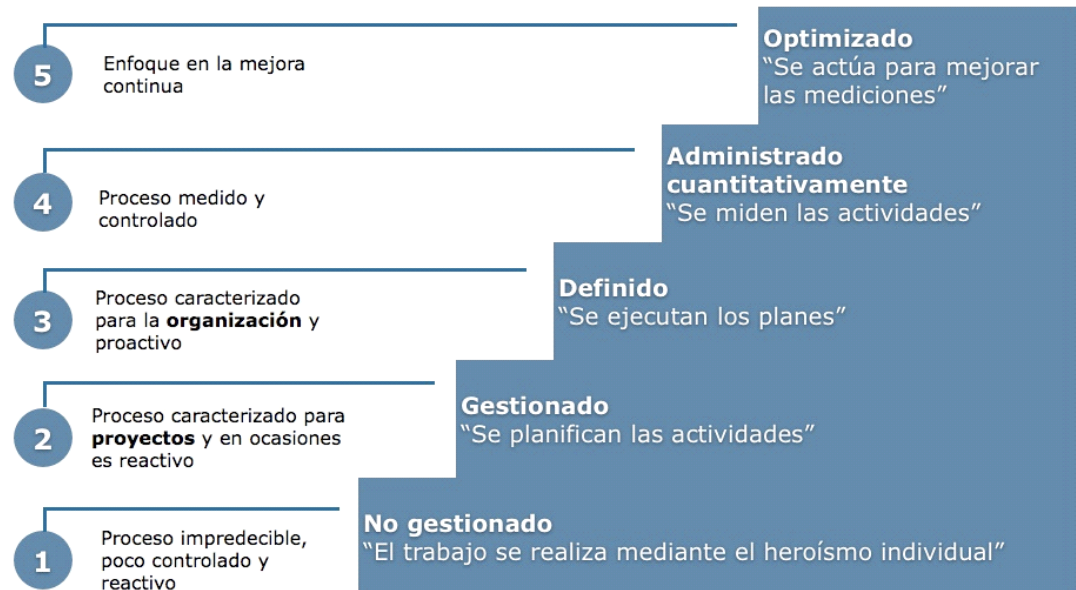


Ilustración 1: Diagrama del modelo CMMI [10].

Se presentaba ante mí el reto de tener que aprender un nuevo lenguaje de programación y su respectivo framework para el desarrollo de aplicaciones web, así como el primer contacto con un sistema de gestión de proyectos dentro de una empresa. Conocer el funcionamiento de Redmine y tratar de añadirle una nueva funcionalidad era trabajo muy interesante que me daría la oportunidad de seguir formándome como profesional en la ingeniería del software. Hay que destacar, que hasta el momento, ningún miembro de la empresa había desarrollado nada en Ruby on Rails, lo cual aumentaba la dificultad de la realización del proyecto al no contar con ayuda alguna dentro de la empresa en cuanto al framework.

1.5 Metodología del proyecto

Para la realización del proyecto se ha seguido la metodología de desarrollo de software en **cascada**. Este modelo destaca se caracteriza por seguir un proceso secuencial de las fases del ciclo de vida de un proyecto software. En este modelo no se realizará la siguiente fase del proyecto hasta que no se ha aprobado la última fase que se estaba realizando [11]. Las fases del proyecto software son:

- **Análisis:** En esta fase se analizan las necesidades del cliente sobre el software para determinar qué objetivos debe cubrir. Se lleva a cabo la especificación de requisitos que contiene la especificación completa de lo que debe hacer el sistema.
- **Diseño:** Si la fase de análisis se completó con éxito, el siguiente paso será formular el diseño técnico del software. El diseño contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada uno de sus componentes, así como la manera en que se relacionan.

- **Implementación:** En esta fase se lleva a cabo la implementación del código fuente del software, así como las pruebas necesarias.
- **Pruebas:** Se comprueba la funcionalidad del sistema mediante el paso de pruebas, tanto del sistema como unitarias. Si existe algún error, se hallará durante el proceso. Se verifica que el sistema cumple todos los requisitos antes de ser entregado al cliente.
- **Integración:** Una vez que se ha verificado el Sistema, la compañía lo pone en producción.
- **Mantenimiento:** En esta fase se realiza una gestión y mantenimiento del sistema para comprobar que el sistema sigue llevando a cabo su función como se deseaba.

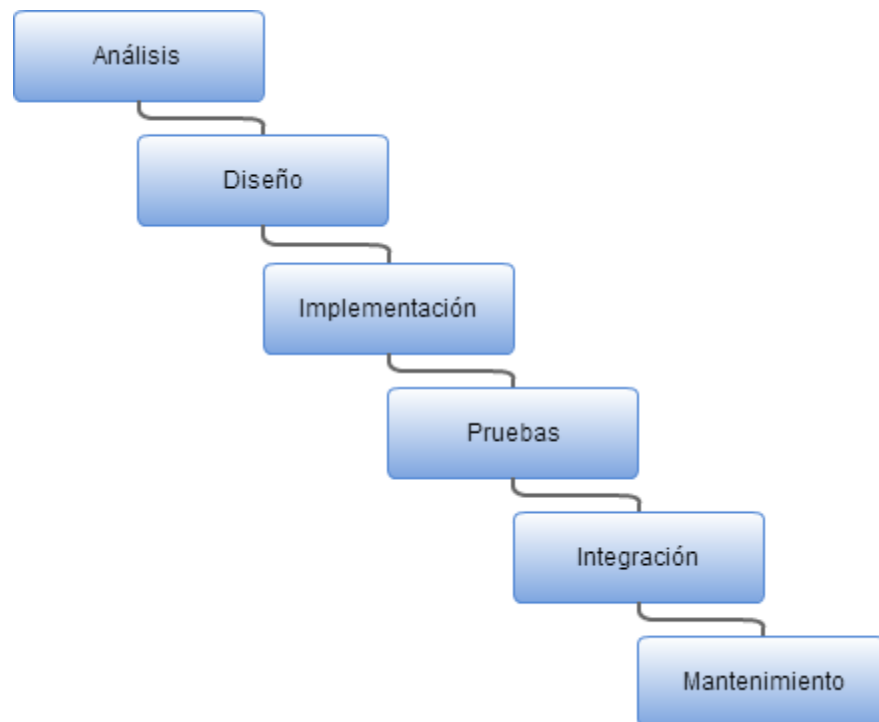


Ilustración 2: Diagrama del modelo en cascada.

1.6 Estructura del documento

A la largo del documento se añadirán los apartados correspondientes a las fases del proyecto software. Cabe destacar, que estos apartados fueron realizados en SATEC como documentos separados, siguiendo la normativa de la empresa. En el documento del plan de proyecto se incluyen los apartados de estudio de viabilidad, en el cual se realizará un estudio de las diferentes tecnologías que se pueden emplear para crear el plug-in y el apartado de planificación del proyecto (ambos documentos no fueron requeridos por SATEC). Además, se incluirán los manuales de usuario e instalación.

Por tanto los apartados que componen el documento del plan de proyecto son los siguientes:

- **Estudio de viabilidad:** En este apartado se realizará el estudio de la propuesta de proyecto para analizar su viabilidad. Aquí se detalla la situación actual del sistema, se realizará un análisis de las herramientas de la competencia y se definirá el sistema a desarrollar realizando un estudio de las alternativas de las que se dispone para el desarrollo.
- **Gestión del proyecto:** Se incluirá la planificación del proyecto con sus respectivos diagramas de Gantt y se indicarán las fechas de entregas. Por otra parte se realizará el coste del proyecto.
- **Análisis:** Aquí se realizará el análisis del proyecto a desarrollar, incluyendo el catálogo de requisitos, los casos de uso y los *mockups* de las interfaces de usuario.
- **Diseño:** En este apartado se definirá la arquitectura del sistema y los planteamientos lógicos que se han dado durante la fase de análisis. Se realizará la creación y explicación de los componentes del sistema y se incluirán los diagramas de clases y el modelo de la base de datos.
- **Plan de pruebas:** El objetivo de este apartado será recoger todas las pruebas posibles para ejecutarlas posteriormente una vez ha concluido la fase de desarrollo. De esta forma se podrá comprobar la calidad del software y que el sistema cumple todos los requisitos marcados por el cliente.
- **Manual de usuario:** Se recogen las indicaciones necesarias para que los usuarios puedan utilizar el plug-in correctamente.
- **Manual de despliegue:** En este apartado se muestran los pasos que se deberán seguir para instalar el plug-in correctamente en el sistema Redmine.

2 | Estudio de viabilidad

2.1 Introducción

El objetivo de este apartado consiste en realizar un estudio del proyecto propuesto para comprobar su viabilidad. Para ello, se realizará un análisis de la situación del sistema Redmine que actualmente utilizan en SATEC y determinará el alcance del proyecto a desarrollar y los usuarios que usarán la herramienta. Por último, se analizarán herramientas similares a Redmine y se realizará un estudio de las diferentes alternativas que se podrán emplear en el desarrollo del Plug-In.

2.2 Descripción del Sistema actual

En este apartado analizaremos el estado actual de la herramienta Redmine. Como hemos comentado en la introducción de este documento, Redmine es una herramienta diseñada para la gestión de proyectos. Incluye varias utilidades para poder realizar el seguimiento de los proyectos, pero carece de alguna que aporte métricas de las peticiones que almacena.

Redmine trabaja en base a peticiones, cuyo ciclo de vida es gestionado a través de la herramienta. Dichas peticiones se clasifican en los siguientes tipos:

- **Funcionalidad:** Se utiliza para introducir en el sistema las funcionalidades a cubrir, y una descripción de las mismas. Normalmente las peticiones tipo “Funcionalidad” serán padre de otras peticiones de más bajo nivel que llevarán a cumplir con esa funcionalidad, principalmente peticiones tipo “Tarea”.
- **Tarea:** Se utiliza para introducir en el sistema peticiones acerca de las labores que se deben realizar en un proyecto y que son las que nos llevarán a conseguir los objetivos. Se recomienda también que la persona encargada de realizar una tarea descomponga está en las subtareas que considere necesario, de cara a una mejor organización del trabajo.
- **Errores:** Se utiliza para introducir en el sistema los errores detectados y las actividades relacionadas con la corrección de los mismos. Los errores pueden descomponerse en subtareas a realizar para solucionarlo. De manera particular para este tipo de peticiones, existe un campo “Origen del error” que ha de ser rellenado para posteriores estudios empleando las métricas y así obtener información sobre las fuentes de errores más habituales.
- **Cambio:** Se utiliza para introducir en el sistema los cambios solicitados. Todo cambio solicitado debe de registrarse, independientemente de que a posteriori se decida aceptarlo o rechazarlo.

- **Mejora:** Se utiliza para introducir en el sistema aquellas peticiones que son propuestas como una mejora del sistema que podría redundar en algún beneficio para el producto.
- **Soporte:** Se utiliza para registrar peticiones de soporte a algún ente externo al equipo de proyecto.
- **Garantía:** Se utiliza para registrar todas las peticiones asociadas a la prestación de la garantía del proyecto. De manera particular para este tipo de peticiones, existe un campo “Origen problema garantía” que ha de ser rellenado para posteriores estudios haciendo uso de las métricas y obtener información de las fuentes más habituales de problemas en periodo de garantía.
- **Peer Review Código:** Se utiliza para registrar los informes de las Peer Reviews realizadas por el Grupo de Peer Review de Código del Área de Aplicaciones de SATEC.
- **Auditoría CM:** Se utiliza para registrar los informes de las Auditorías de Gestión de la Configuración realizadas por el Grupo de Auditorías de Gestión de la Configuración del Área de Aplicaciones de SATEC.
- Todos los datos de estas peticiones (número de peticiones, estado de la petición, horas realizadas, etc.) son los necesarios para poder obtener métricas, de tal forma que estas se procesen y presenten los datos obtenidos de forma gráfica y tabular.
- En cuanto a los usuarios que utilizan la herramienta Redmine, existen dos **perfiles de usuario** definidos:
- **Administradores:** El usuario administrador tiene control total de la aplicación Redmine y su labor es gestionar el sistema. Dispone de un panel a través del cual gestiona los principales parámetros de la plataforma, como la gestión de usuarios, grupos y perfiles. La gestión de usuarios se realiza por medio de un listado a través del cual se puede buscar a un usuario mediante un buscador, y así poder modificar sus preferencias, añadir o eliminar nuevos usuarios.
- **Usuarios:** Dentro de este grupo se pueden diferenciar cuatro tipos:
 - Responsable de proyecto: Es el encargado de gestionar el proyecto. Tiene capacidad para realizar todas las acciones que permite la herramienta sobre el proyecto en el cual es responsable.
 - Desarrollador: Perfil con un conjunto limitado de permisos entre los que destacan: gestión de mensajes propios en los foros, ver y añadir documentos, gestionar ficheros, ver el diagrama de Gantt del proyecto, gestionar las peticiones, comentar noticias, consultar el repositorio y las peticiones relacionadas, gestión del tiempo dedicado a las peticiones y acceso de escritura a la wiki.

- No miembro: Perfil que tiene todo usuario autenticado en el sistema en un proyecto en el cual no ha sido dado de alta.
- Anónimo: Perfil no utilizado en la instalación de SATEC que tiene toda persona que accede al sistema sin autenticarse. Como en nuestra instalación es obligatoria la autenticación, este perfil no se aplica.

2.3 Definición del Sistema a desarrollar

Una vez descrita la situación actual de Redmine, procederemos a definir el alcance del sistema detallando cómo afectará a la aplicación de Redmine, así como el tipo de usuarios que hacen uso del Plug-In.

2.3.1 Alcance

El sistema a desarrollar consiste en un Plug-In (empleando el *framework* Ruby on Rails) para la aplicación Redmine que permita representar una serie de métricas de forma **gráfica** y **tabular**. Dichas métricas podrán ser creadas, editadas y borradas solamente por los usuarios a los que se les haya asignado esos privilegios (ver apartado 3.2). A continuación se detallan los datos que formarán las métricas de nuestro Plug-In:

- **Título**: Nombre de la métrica.
- **Consulta SQL**: Consulta sobre la base de datos de Redmine con el objetivo de obtener los datos a representar en la tabla y el gráfico.
- **Tipo de gráfico**: Como ya hemos comentado en la introducción de este documento, para la representación de estos gráficos se utilizará un *framework* llamado Chartkick, el cual nos permite seleccionar entre gráficos de columnas, de líneas, de tartas o de áreas. Adicionalmente, se permitirá que se establezca que datos representarán los valores y las etiquetas de los gráficos.

Una vez explicado el contenido de las métricas, se diferenciarán los dos tipos de métricas que existen dependiendo del dominio que abarquen estas:

- **Generales**: Se trata de métricas que se aplican a todo Redmine, es decir obtienen datos de todos los proyectos alojados en la aplicación.
- **De proyecto**: El dominio de estas métricas es el propio proyecto en sí mismo, por tanto solo mostrarán datos del proyecto en cuestión.

Para poder visualizar un tipo de métrica de los mencionados anteriormente, como para poder configurarlas, es necesario que el usuario tenga una serie de permisos. A continuación se describen los tipos de permisos que se pueden asignar a los usuarios:

- **Permisos de configuración de métricas:** Estos permisos hacen referencia a la creación, edición y eliminación de métricas en Redmine. Como ya hemos comentado, existen dos tipos de métricas (generales y de proyecto) para las cuales se diferenciarán también sus respectivos permisos, que pueden ser:
 - **Crear métrica general:** Este permiso otorga al usuario el poder de crear una nueva métrica general.
 - **Editar métrica general:** Determina que el usuario puede editar una métrica general ya existente en la aplicación Redmine.
 - **Eliminar métrica general:** Otorga al usuario el poder de eliminar una métrica general.
 - **Crear métrica de proyecto:** Este permiso otorga al usuario el poder de crear una nueva métrica de proyecto.
 - **Editar métrica de proyecto:** Determina que el usuario puede editar una métrica de proyecto ya existente en la aplicación Redmine.
 - **Eliminar métrica de proyecto:** Otorga al usuario el poder de eliminar una métrica de proyecto.
- **Permisos de visualización de métricas:** También existen dos tipos de permisos relacionados con el tipo de métrica que se desea visualizar:
 - **Ver métricas generales:** El usuario que tenga este permiso podrá visualizar las métricas generales de Redmine.
 - **Ver métricas de proyecto:** El usuario que tenga este permiso podrá visualizar las métricas del proyecto al que están asignados.

2.3.2 Identificación de los usuarios participantes

Para poder utilizar el plug-in los usuarios necesitarán una serie de permisos según las acciones que se deseen realizar. Por ello solo se distinguen dos tipos de usuarios:

- **Usuarios con permisos:** Se tratan de usuarios de Redmine a los que el administrador les ha concedido determinados permisos para poder realizar acciones sobre de métricas, como pueden ser crear, editar, eliminar y visualizar métricas.
- **Usuarios sin permisos:** Resto de usuarios de Redmine. No tienen privilegios para poder realizar ninguna acción con las métricas. Para que un usuario pueda realizar acciones sobre las métricas el administrador podrá asignarle ciertos permisos.

2.4 Estudio de la situación actual

La tendencia creciente al trabajo con equipos de trabajo remotos, ha provocado que las empresas del sector de la ingeniería Software, tiendan a invertir una cantidad considerable de dinero por la licencia de herramientas de gestión de proyectos potentes. Por todo ello, hay un auge de herramientas para trabajar online de forma colaborativa, que aportan comunicación instantánea y permiten trabajar en un entorno de información compartida y actualizada. Las principales funcionalidades que aportan estas herramientas son:

- Seguimiento de tareas.
- Seguimiento de plazos.
- Seguimiento de errores.
- Seguimiento de costes.
- Gestión de la relación con clientes.
- Intercambio de información (Wiki).
- Control de versiones (Github o SVN).
- Mensajería instantánea y conferencias en tiempo real.

Este conjunto de funcionalidades hacen que las herramientas colaborativas para la gestión de proyectos sean muy interesantes, no solo para el equipo de desarrollo, sino también para la gerencia de la empresa. Normalmente, las herramientas que requieren el pago de una licencia para poder utilizarlas, cubren más funcionalidades que las herramientas gratuitas. Dependiendo del tamaño de la empresa o del presupuesto del que disponen, se opta por hacer con una de estas herramientas que requieren licencia. A continuación, se exponen dos herramientas de gestión de proyectos que requieren licencia para su uso:

2.4.1 Asamblea

Esta herramienta colaborativa ofrece componentes para la administración de proyectos de desarrollo software. Permite gestionar y trabajar efectivamente en equipo, permitiendo el seguimiento de tareas y plazos, permitiendo realizar Scrum (reparto de tareas), así como la integración con Subversion para poder realizar un control de versiones de los archivos [12]. El elemento clave para la gestión de tareas, es el sistema de tickets que son asignados a cada miembro del equipo. Cada ticket incorpora información detallada como el nivel de complejidad y las horas de trabajo. Destaca por sus funcionalidades de Reporting, que incluyen informes muy útiles para gestionar plazos y recursos, e informes de errores y quejas de clientes.

Mantiene todos los documentos asociados con un proyecto en un solo lugar para facilitar la comunicación entre el equipo trabajando en el proyecto. Además, incorpora herramientas de intercambio de información como la Wiki y la mensajería.

El coste de licencia mensual para 30 usuarios es de 49\$, mientras que para 50 usuarios es de 99\$.



Ilustración 3: Logo de Asamblea.

2.4.2 Redbooth

Antiguamente conocida como Teambox, contiene en una única herramienta los mecanismos necesarios para que su equipo pueda organizar, comunicar, compartir y trabajar de forma conjunta en tiempo real, maximizando la productividad y el rendimiento para su empresa.

Permite la creación y gestión de tareas, para organizar los proyectos asignando responsabilidades para que el equipo trabaje de manera más eficiente, aplicando la metodología Scrum. Genera informes de los resultados de las tareas, mostrando información como cuales son las tareas urgentes o cuales han sufrido un retraso frente al plazo de finalización previsto. Además incluye un calendario de tareas y un diagrama de Gantt para poder realizar un seguimiento de forma más rápida visualmente [13].

Contiene componentes para la gestión de archivos, los cuales se pueden compartir entre el equipo de trabajo. Además, incluye el servicio de almacenamiento de archivos Cloud.

En las últimas versiones han incorporado una herramienta chat, así como la posibilidad de realizar videoconferencias en HD, lo cual permite mejorar la comunicación entre los miembros del equipo, algo que ayudará a reducir el número de correos que se intercambian también sobre un proyecto. Además, se cuenta con una aplicación móvil del servicio, la cual permite gestionar los proyectos y recibir notificaciones a nuestro dispositivo.

Redbooth cuenta con una opción gratuita de hasta cinco usuarios, con un máximo de cinco proyectos y 5 GB de almacenamiento en la nube. Sin embargo, el coste de las licencias son un poco más caras que en el caso de Assembla. La versión Pro parte de los 5\$ por usuario y mes que tiene espacio y proyectos ilimitados además de las nuevas funcionalidades. 20 Usuarios 99\$ mes y para 50 tarifa a medida negociando con ventas.



Ilustración 4: Logo de Redbooth.

2.5 Estudio de alternativas de solución

En este apartado, se realizará un estudio de las alternativas para generar gráficos en el Plug-In, así como para generar documentos en formato PDF que recojan los resultados de las métricas.

2.5.1 Representación de gráficos

A continuación, se recogen cuatro librerías que sirven para generar gráficos.

2.5.1.1 Morris.js

Morris.js es una librería simple de código abierto que permite generar gráficos simples en JavaScript [14]. Los requisitos mínimos son: jQuery 1.7 y Raphael.js 2.0. Para representar los gráficos, requiere que los datos estén recogidos en un array en el fichero donde está declara la función de jQuery. Ofrece pocas opciones de configuración de representación de los gráficos.

2.5.1.2 Google Charts

Se trata de una potente librería gratuita que permite representar gráficos en JavaScript con múltiples configuraciones [15]. Ofrece gráficos muy completos con una amplia variedad. Es una de las herramientas más utilizadas para representar gráficos debido a su sencillez y buena representación. Al igual que Morris.js requiere que los datos estén recogidos en un array en el fichero donde está declara la función de jQuery. Para su correcto funcionamiento, serán necesarias las siguientes librerías: Google JSAPI API, la librería de visualización de Google y la propia librería del gráfico seleccionado.

2.5.1.3 Highcharts

Highcharts es una librería muy similar a Google Charts, ofreciendo gráficos muy completos. La mayor ventaja que encontramos en esta librería es la posibilidad de exportar el gráfico generado a ficheros en formato PNG, JPG, PDF en la parte cliente [16]. El requisito mínimo para su funcionamiento es la versión de jQuery 1.3.2 y deberá incorporar una única librería de Highcharts. Necesita que los datos del gráfico estén recogidos en array o en un mapa para poder realizar la representación del gráfico.

2.5.1.4 Chartkick

Chartkick es una herramienta que permite representar gráficos en JavaScript con solamente una única línea de código en Ruby [17]. Su compatibilidad con Ruby on Rails es lo que hace a esta herramienta más interesante frente al resto que se han comentado anteriormente. Para llevar a cabo la representación de los gráficos, permite elegir entre utilizar las librerías de Google Charts o Highcharts, las cuales ofrecen una amplia variedad de configuraciones posibles. El requisito mínimo para hacer uso de esta herramienta es la versión 2.3 de Rails y se deberá instalar la gema correspondiente a Chartkick, así como incorporar la librería JavaScript de representación de gráficos de Google Charts o Highcharts. La estructura de datos que requiere para representar los gráficos pueden ser un array o un mapa.

2.5.1.5 Gruff

Gruff permite generar gráficos en código Ruby, con la particularidad de que se permite su exportación en formato PNG en el lado servidor [18]. Los gráficos que genera son simples y no ofrecen tantas alternativas de configuración. Para su uso, bastará con instalar la gema correspondiente a la librería.

2.5.2 Generación de documentos PDF

Las librerías encontradas para la generación de documentos en formato PDF son las siguientes.

2.5.2.1 Prawn

Prawn se trata de una librería para generación de documentos PDF con código Ruby [19]. Con esta herramienta se pueden generar documentos con un diseño sencillo de forma rápida. Incluye una extensa documentación de todas las posibilidades de diseño disponibles. Permite añadir imágenes en formato PNG y JPG a los documentos de forma rápida y sencilla. Incorpora herramientas de seguridad como la encriptación del documento o protección con contraseña. La instalación de Prawn se realiza a través de su gema y los requisitos mínimos de esta librería es la versión de Ruby 1.7.

2.5.2.2 PDFKit

Esta herramienta permite transformar el texto plano HTML+CSS en un documento PDF, mediante la utilidad *wkhtmltopdf* [20]. El principal inconveniente de esta herramienta lo encontramos ante la necesidad de insertar los gráficos generados, ya que existen problemas a la hora de insertar imágenes o gráficos de JavaScript. Para usar esta herramienta se requiere instalar su gema correspondiente.

2.5.2.3 Wicked PDF

Al igual que PDFKit, esta herramienta transforma el documento HTML para generar un PDF con la utilidad *wkhtmltopdf* [21]. Tampoco ofrece soluciones a la hora de insertar imágenes o gráficos de JavaScript.

2.5.3 Selección de la solución

Una vez analizadas las alternativas, se procederá a elegir la solución que se ha creído más conveniente para el desarrollo del componente.

2.5.3.1 Representación de gráficos

En un principio, Highcharts se postuló como la alternativa más adecuada a las necesidades del proyecto, sin embargo tenía dos inconvenientes que la descartaron: no permitía la generación de gráficos a través de Ruby (únicamente permitía a través de jQuery) y la exportación del gráfico se realizaba en el lado cliente sin poder modificar el documento PDF, lo cual no permitía añadir la tabla de resultados de la métrica.

Por ello, se ha elegido **Chartkick** como herramienta para generar gráficos en la web de Redmine, utilizando la librería de Google Charts, la cual se ha considerado que el estilo encajaba mejor en Redmine que la de Highcharts. El factor determinante de esta elección, ha sido la posibilidad de generar los gráficos en una sola línea de Ruby y el uso fácil de los datos en Ruby on Rails, ya que se podrán obtener los datos directamente de la ejecución de la consulta SQL para su posterior representación sin tener que realizar ninguna modificación compleja en la estructura de datos.

Además, dado que se requiere que en los ficheros PDF se muestre un gráfico de los resultados de las métricas, se ha elegido utilizar la herramienta **Gruff** para generar un fichero PNG en el lado servidor. Esto nos permitirá utilizar la imagen del gráfico en la generación del documento PDF. También se ha elegido esta alternativa por su similitud en la generación de gráficos con la herramienta Chartkick, ya que utilizan las mismas estructuras de datos.

2.5.3.2 Generación de documentos PDF

Para la generación de documentos PDF se ha elegido la herramienta **Prawn**, claramente por ser la única que se adaptaba mejor a las necesidades del proyecto. La posibilidad de insertar imágenes ha sido clave a la hora de tomar la decisión, así como la gran cantidad de documentación que ofrecen en su manual, lo cual hace que la labor de generar un documento PDF no sea complicada en Ruby.

3 | Gestión de proyecto

3.1 Planificación

En este apartado se detallará la planificación de las tareas en las que se descompone el proyecto, cuya realización seguirá la metodología en cascada. Debido al gran número de tareas y de su propagación en el tiempo, se mostrará primero el diagrama de Gantt de las tareas principales y después se profundizará en cada una de ellas mostrando su parte del diagrama de Gantt. A continuación se muestra el listado de tareas principales del proyecto y su respectivo diagrama de Gantt:

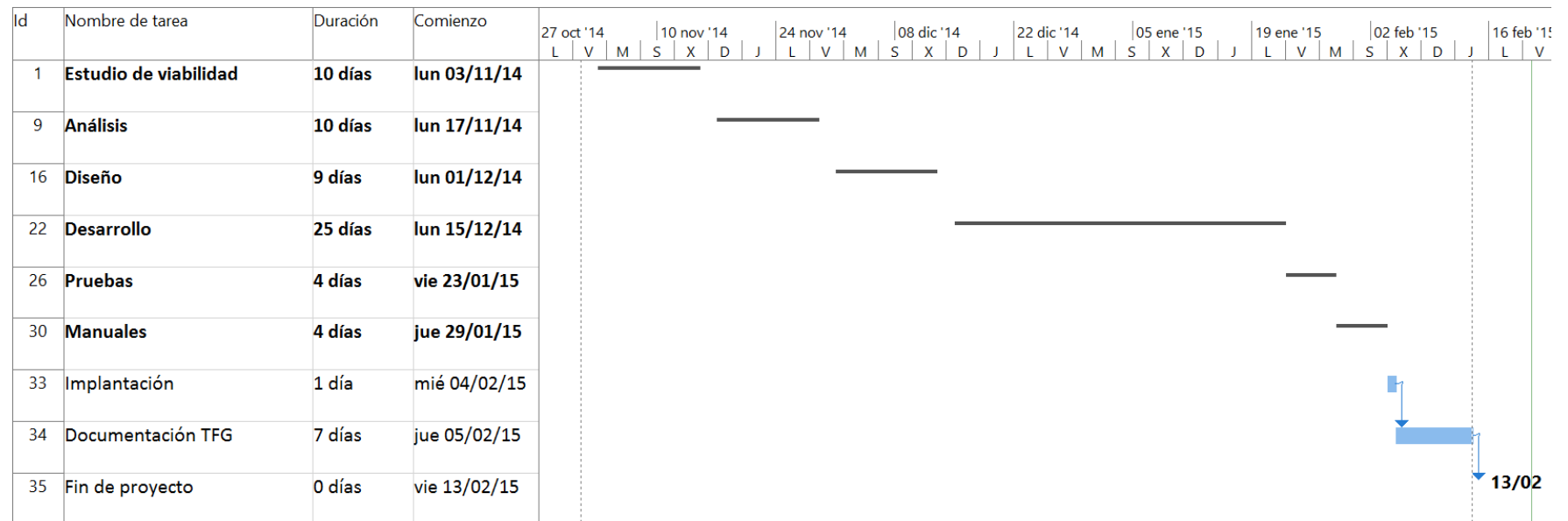


Ilustración 5: Diagrama de Gantt del proyecto.

Como podemos observar en la imagen anterior, el proyecto dará comienzo el día 3 de Noviembre del año 2014 y tiene fijada su fecha de finalización el día 13 de Febrero del año 2014, es decir, el proyecto tendrá una duración aproximada de tres meses.

3.1.1 Estudio de viabilidad

La fase de realización del estudio de viabilidad del proyecto tendrá una duración de 10 días. Se realizará una investigación sobre cómo se utiliza la herramienta Redmine y como está desarrollada, así como el estudio de las tecnologías que se podrán aplicar al desarrollo del componente. La información será recogida en el apartado de estudio de viabilidad del documento de plan de proyecto. En la siguiente imagen se puede observar las tareas en las que se descompone esta fase del proyecto y su respectivo diagrama de Gantt:

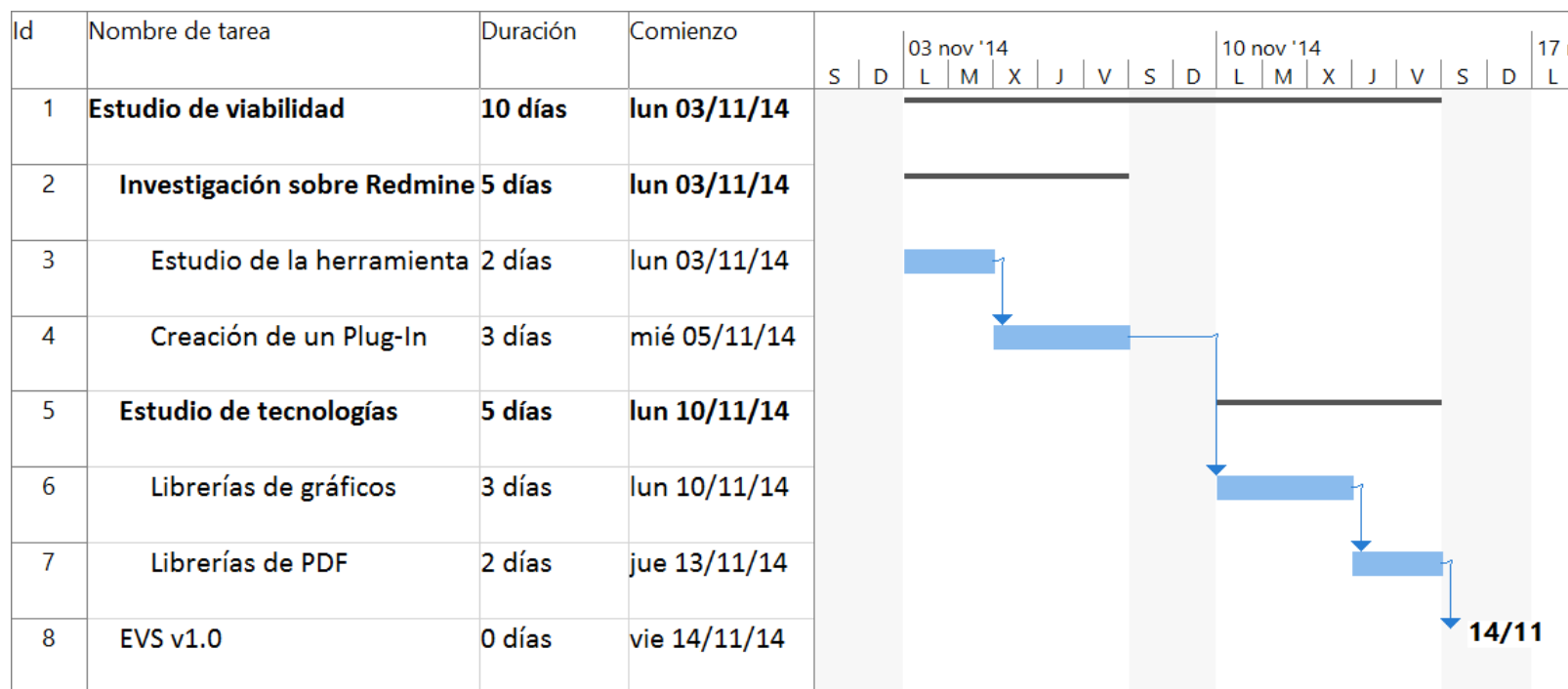


Ilustración 6: Diagrama de Gantt del estudio de viabilidad.

3.1.2 Análisis

La fase de análisis tendrá una duración de 10 días. El objetivo de esta fase será el de realizar un apartado del documento del plan de proyecto que recoja el catálogo de requisitos (requisitos de usuario, funcionales y no funcionales), así como los subsistemas por los que estará compuesto el Plug-In, sus casos de uso correspondientes, el modelo lógico de datos y las interfaces de usuario. A continuación, se muestra el diagrama de Gantt asociado a esta fase del proyecto:

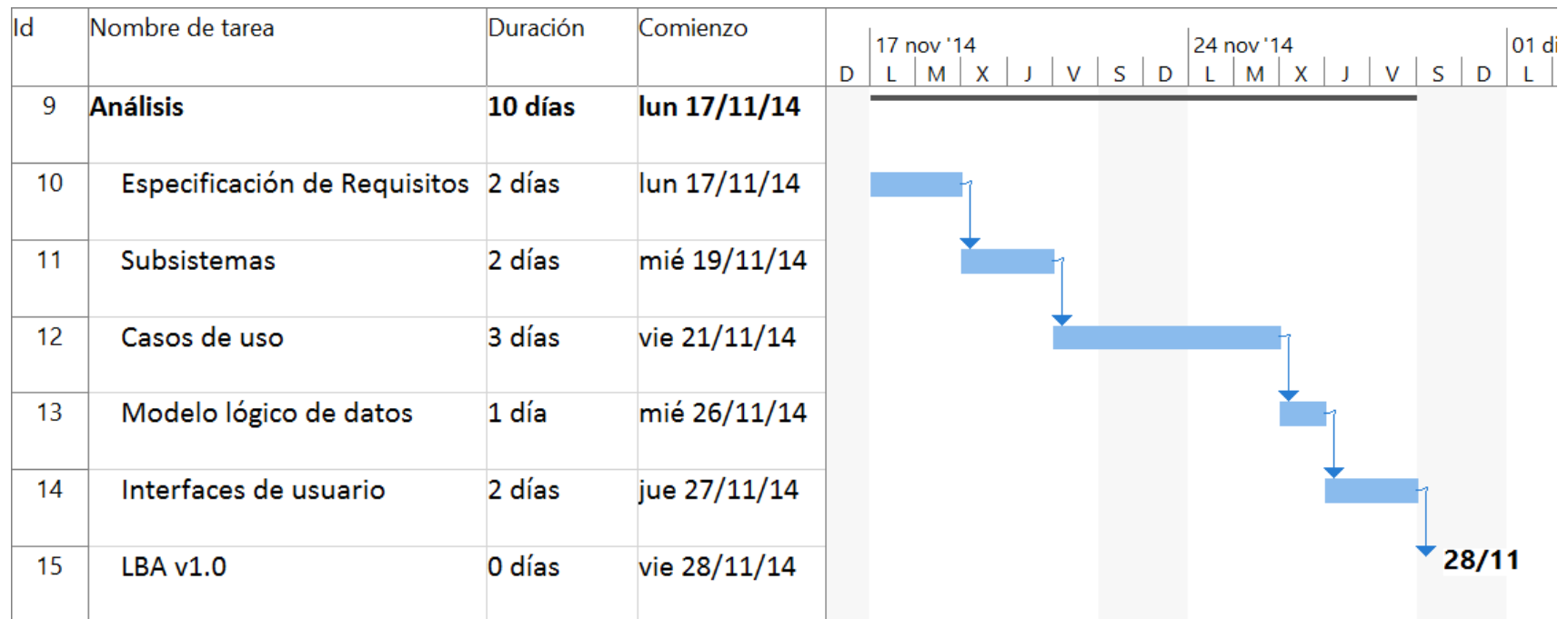


Ilustración 7: Diagrama de la fase de análisis.

3.1.3 Diseño

Esta fase del proyecto tendrá una duración de 10 días. En ella se llevará a cabo las tareas necesarias para cumplimentar el apartado de diseño del plan de proyecto. En la siguiente imagen se puede observar las tareas en las que se descompone la fase de diseño y su respectivo diagrama de Gantt:

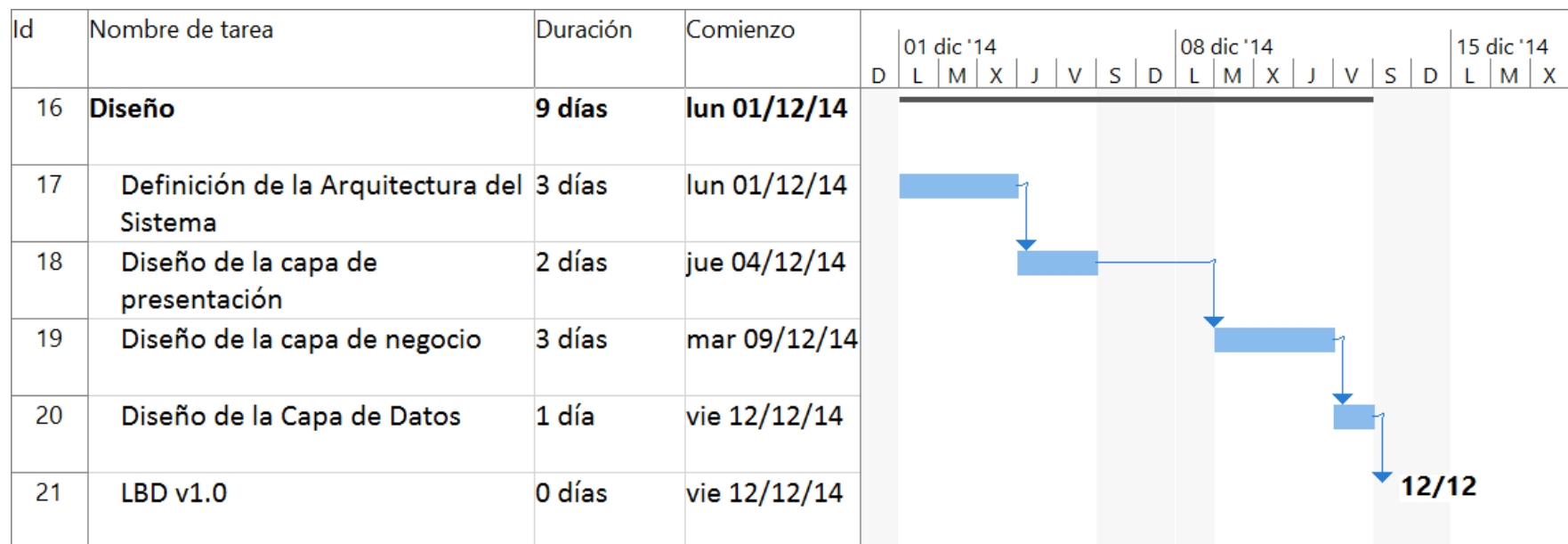


Ilustración 8: Diagrama de Gantt de la fase de diseño.

3.1.4 Desarrollo

La fase de desarrollo será la más extensa del proyecto, ya que tendrá una duración estimada de 25 días. En la siguiente imagen se puede observar las tareas en las que se descompone la fase de desarrollo y su respectivo diagrama de Gantt:

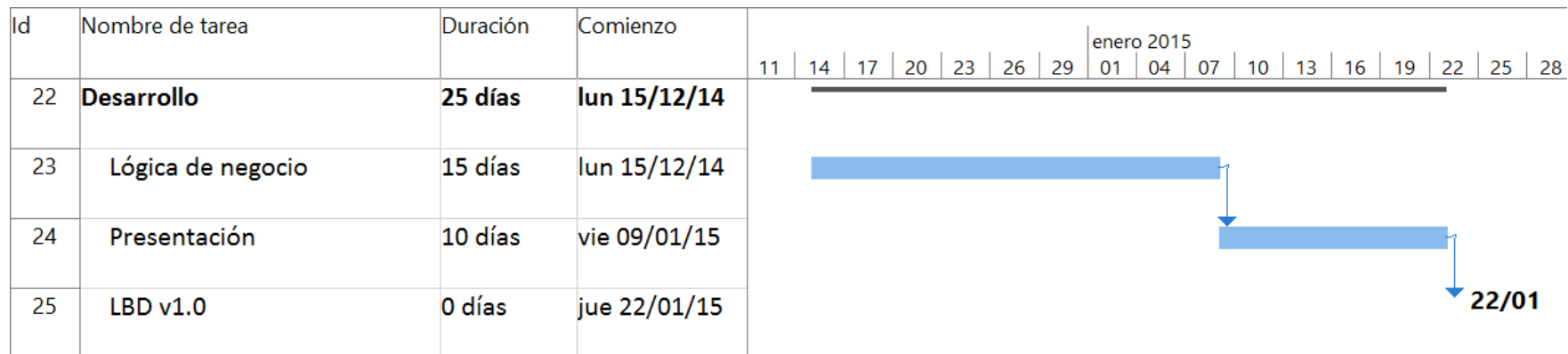


Ilustración 9: Diagrama de Gantt de la fase de desarrollo.

3.1.5 Pruebas

Una vez terminada la fase de diseño del componente, se realizará el apartado de plan de pruebas y posteriormente se procederá a ejecutar todas las pruebas para que verifique el software cumple lo prometido. A continuación, se muestra el diagrama de Gantt correspondiente:

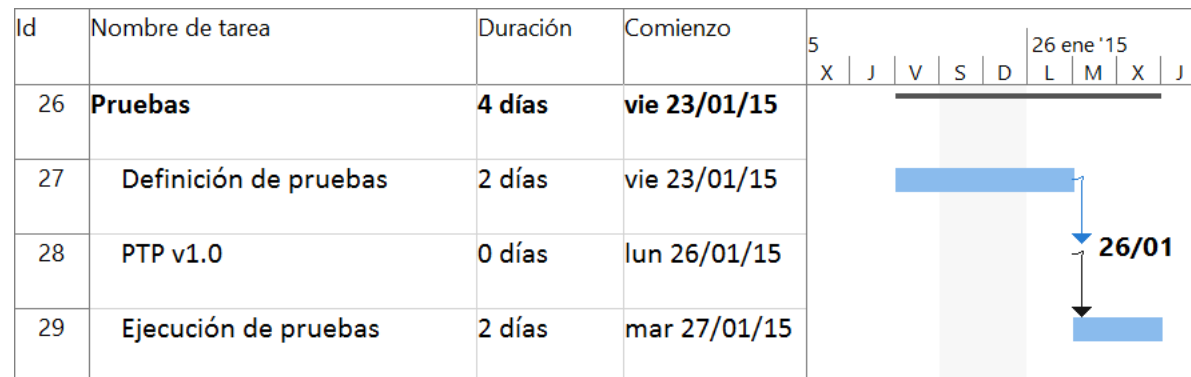


Ilustración 10: Diagrama de Gantt de la fase de pruebas.

3.1.6 Resto del proyecto

Terminada la fase de desarrollo del Plug-In, se llevarán a cabo las tareas de elaboración de manuales. Una vez comprobado, el Plug-In se implantará en el sistema Redmine de SATEC para que los usuarios puedan utilizarlo. La última tarea a realizar para dar por finalizado el trabajo de fin de grado, será ajustar la documentación a la normativa de la universidad. . A continuación, se muestra el diagrama de Gantt asociado a las últimas tareas del proyecto:

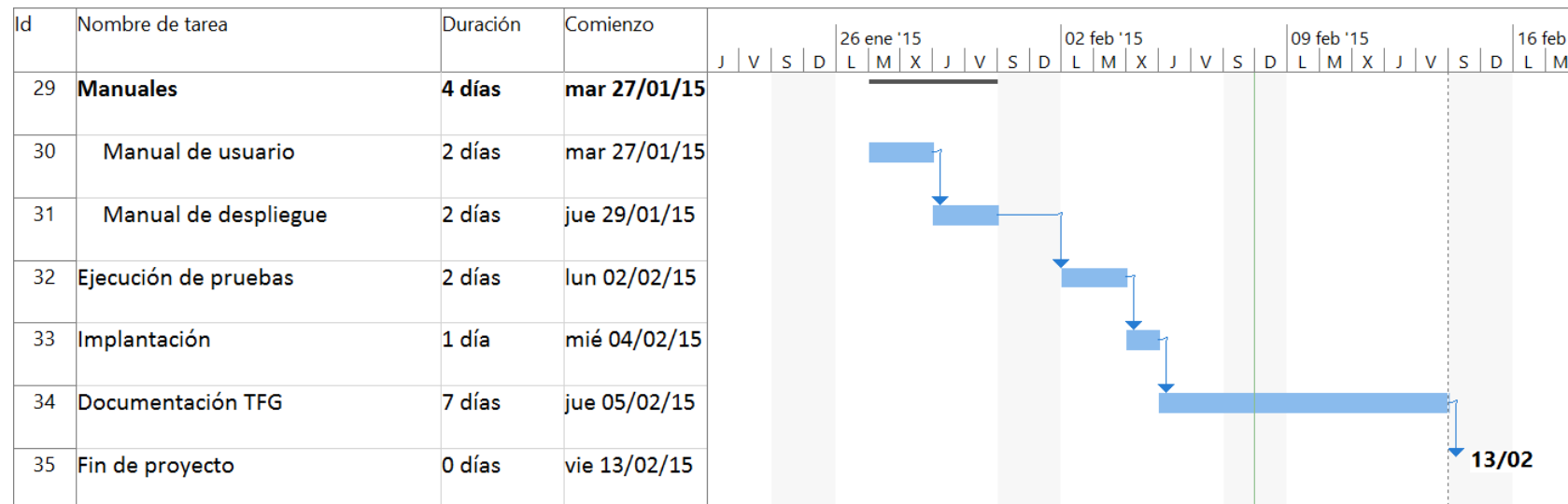


Ilustración 11: Diagrama de Gantt del resto del proyecto.

3.1.7 Hitos

Cabe destacar las fechas de entrega de los documentos y del ćdigo que corresponden con las ĺneas base, aś como la fecha de fin de proyecto que se corresponde con la puesta en producci3n del Plug-In:

Hito	Fecha
EVS v1.0	14/11/2014
LBA v1.0	28/11/2014
LBD v1.0	12/12/2014
LBE v1.0	22/01/2014
PTP v1.0	26/01/2014
Fin de proyecto	13/02/2014

3.2 Costes

En este apartado, se detallar_ coste del proyecto. Hay que se_alar, que sobre este proyecto se establecer_ un **margen de beneficio** del **20%**.

En la siguiente tabla se recoge el coste del personal que ha realizado el proyecto, el cual tendr_ una duraci3n de **70 d́as** en los cuales la jornada laboral dura **6 horas**. Cabe destacar que los puestos no son m_ s que roles que he adoptado durante el ciclo de vida del proyecto.

Personal	Coste/Hora	Horas dedicadas	Importe
Jefe de proyecto	20,00 €	150	3.000,00 €
Analista	15,00 €	60	900,00 €
Dise_ador	15,00 €	54	810,00 €
Programador	10,00 €	198	1.980,00 €
Total			6.690,00 €

Cabe destacar, que en el coste por hora de cada trabajador es corresponde al coste bruto (impuestos incluidos). Adem_ s, en el coste se incluyen los gatos que genera un trabajador a la hora en cuanto a servicios como internet, luz, agua y gas.

3.2.1 Equipo informático

El coste del equipo informático utilizado por el personal es el siguiente:

Nombre	Cantidad	Precio Unidad	Total
HP EliteBook 820 G1	4	1.148,29 €	4.593,16 €
Ratón HP Z4000	4	30,00 €	120,00 €
Coste total (Sin IVA)			4.713,16 €

3.2.2 Licencias Software

Las licencias software necesarias para llevar a cabo la labor de los trabajadores han tenido el siguiente coste:

Nombre	Cantidad	Precio Unidad	Total
Windows 7 Professional 64x	4	192,70 €	770,80 €
Microsoft Office 2013	4	69,90 €	279,60 €
Enterprise Architect	4	339,00 €	1.356,00 €
Coste total (Sin IVA)			2.406,40 €

3.2.3 Material fungible

El coste del material fungible necesario se muestra a continuación:

Nombre	Cantidad	Precio Unidad	Total
Bolígrafos Caja 10 Uds	4	4,20 €	16,80 €
Cuaderno 100 hojas	4	1,30 €	5,20 €
Coste total (Sin IVA)			22,00 €

3.2.4 Coste total del proyecto

Además de los costes calculados anteriormente, se deberá añadir los **costes indirectos** del proyecto como son luz, agua, internet, etc. Por lo tanto, el coste total del proyecto es el siguiente:

Descripción	Coste
Personal	6.690,00 €
Equipos	4.713,16 €
Licencias Software	2.406,40 €
Material fungible	22,00 €



Coste total	13.831,56 €
M.B. 20%	16.597,87 €
IVA 21%	20.083,43 €

4 | Análisis

4.1 Introducción

A continuación, se incluye el análisis del proyecto '**Métricas Redmine**' cuyo objetivo es la especificación detallada del sistema a desarrollar que satisfaga las necesidades extraídas del cliente y que sirva de base para el posterior diseño.

En este proyecto, se procederá al desarrollo de un **Plug-In** para la aplicación **Redmine**. Redmine es una herramienta escrita usando el *framework* **Ruby on Rails**, diseñada para la gestión de proyectos que incluye un sistema de seguimiento de incidencias. Otras utilidades que incluye Redmine son: calendario de actividades, diagramas de Gantt para la representación visual de la línea del tiempo de los proyectos, wiki, foro, visor del repositorio de control de versiones, RSS, control de flujo de trabajo basado en roles, integración con correo electrónico, etcétera.

El Plug-In a desarrollar se encargará de representar, mediante gráficos y tablas, una serie de métricas acerca de los proyectos creados en Redmine. Dichas métricas serán establecidas por un usuario con permisos. Para la representación de estos gráficos se utilizará un *framework* llamado **Chartkick**, el cual permite representar gráficos de **Java Script**.

En los siguientes apartados se detallarán los objetivos del proyecto, así como la definición del sistema, delimitando el alcance del mismo y los permisos de los usuarios involucrados. Además, se detallarán los requisitos del sistema para facilitar el diseño y la codificación, los casos de uso con el sistema y se establecerá el diagrama de entidades.

4.1.1 Objetivos y restricciones

El objetivo principal de este proyecto es poder ofrecer a los usuarios autorizados de la herramienta Redmine la posibilidad de obtener datos de los proyectos de forma visual, para una mejor captación de la información. Hay que destacar que el Plug-In se desarrollará para uso interno en la empresa **Satec**.

A continuación se especificarán los **stakeholders** implicados en el proyecto. Debe señalarse que al tratarse de un proyecto interno, el rol de cliente será realizado por el jefe de proyecto. Los stakeholders del proyecto son:

- **Cliente:** Con él se acuerdan los requisitos que tendrá el sistema, ya que es quien dará la aprobación del producto.
- **Administradores de Redmine:** Su función es la de administrar la aplicación de Redmine, por lo que también serán los encargados de administrar el Plug-In de métricas.
- **Usuarios:** Este grupo lo conforman todos los empleados de Satec que hacen uso de la herramienta Redmine para gestionar sus proyectos.

4.2 Cat́logo de Requisitos

En este apartado se recoge la especificaci3n de requisitos del Plug-In de m3tricas. El cat́logo de requisitos se ha dividido en tres sub-apartados dependiendo del tipo de requisito: Requisitos de usuario, funcionales y no funcionales.

4.2.1 Requisitos de usuario

Los requisitos de usuario recogen el conjunto de condiciones o capacidades que un usuario necesita para poder resolver un problema o lograr un objetivo. A continuaci3n se muestra una tabla con los requisitos de usuario para el Plug-In de m3tricas:

ID	Descripci3n	Criticidad
RU-001	El usuario debe conocer el lenguaje SQL para poder crear m3tricas.	Importante

4.2.2 Requisitos Funcionales

Este tipo de requisitos representan descripciones del sistema o que el sistema debe ser capaz de realizar. A continuaci3n se muestra una tabla con los requisitos funcionales del Plug-In de m3tricas:

ID	Descripci3n	Criticidad
RF-001	El usuario deber1 tener los permisos necesarios para poder realizar las acciones que ofrece el plug-in.	Importante
RF-002	Un usuario podr1 crear m3tricas nuevas si tiene permiso.	Importante
RF-003	Un usuario podr1 editar m3tricas ya existentes si tiene permiso.	Importante
RF-004	Un usuario podr1 eliminar m3tricas si tiene permiso.	Importante

RF-005	El usuario deberá rellenar un formulario para crear una métrica.	Importante
RF-006	El sistema deberá mostrar un mensaje de error si falta algún campo obligatorio por rellenar en el formulario de métricas.	Importante
RF-007	El sistema deberá mostrar un mensaje de error si algún campo se ha rellenado de forma incorrecta en el formulario de métricas.	Importante
RF-008	El sistema deberá mostrar un mensaje de que la métrica ha sido creada con éxito.	Importante
RF-009	El sistema deberá mostrar un mensaje de que la métrica ha sido editada con éxito.	Importante
RF-010	El sistema deberá mostrar un mensaje de confirmación de eliminación de la métrica.	Importante
RF-011	El sistema deberá mostrar un mensaje de que la métrica ha sido eliminada con éxito.	Importante
RF-012	Un usuario asociado a un proyecto podrá ver las métricas asociadas a dicho proyecto si tiene permisos para ello.	Importante

RF-013	Un usuario podrá visualizar métricas generales de todos los proyectos de Redmine si tiene permisos para ello.	Importante
RF-014	El sistema deberá mostrar en la parte de visualización de métricas los datos obtenidos por la consulta de una métrica en una tabla.	Importante
RF-015	El sistema deberá mostrar en la parte de visualización de métricas los datos obtenidos por la consulta de una métrica en una gráfica, cuyo tipo fue indicado en la creación de la métrica.	Importante
RF-016	El sistema deberá mostrar un mensaje de error al usuario si se produce un error al ejecutar la consulta de la métrica.	Importante
RF-017	El sistema deberá mostrar una nueva opción de menú "Configuración de métricas" dentro del menú "Administración", la cual solo será visible por los usuarios que tengan permisos asignados.	Importante
RF-018	El sistema deberá mostrar la opción de crear una nueva métrica dentro del menú "Configuración de métricas" que será visible por los usuarios que tengan permisos asignados.	Importante
RF-019	El sistema deberá mostrar la opción de editar una métrica dentro del menú "Configuración de métricas", que será visible por los usuarios que tengan permisos asignados.	Importante

RF-020	El sistema deberá mostrar la opción de eliminar una métrica dentro del menú “Configuración de métricas”, que será visible por los usuarios que tengan permisos asignados.	Importante
RF-021	El sistema deberá mostrar en el menú superior una nueva pestaña llamada “Métricas generales”, que será visible por los usuarios que tengan permisos asignados.	Importante
RF-022	El sistema deberá mostrar un listado de las métricas generales almacenadas en la base datos en la pestaña “Métricas generales” del menú superior.	Importante
RF-023	El sistema deberá mostrar en el menú de proyecto una nueva pestaña llamada “Métricas”, que solo será visible por los usuarios que tengan acceso a dicho proyecto y tengan permisos de visualización.	Importante
RF-024	El sistema deberá mostrar un listado de las métricas del proyecto al que se ha accedido a través de la pestaña “Métricas” del menú del proyecto.	Importante
RF-025	El sistema ofrecerá al usuario la opción de poder exportar los resultados obtenidos en forma tabular de la métrica que puede visualizar en un fichero con formato CSV.	Secundario
RF-026	El sistema ofrecerá al usuario la opción de poder exportar los resultados obtenidos de la métrica que puede visualizar en un fichero con formato PDF.	Importante

4.2.3 Requisitos No Funcionales

Estos requisitos especifican criterios que pueden usarse para juzgar la operación de un sistema. El conjunto de requisitos no funcionales se recoge en la siguiente tabla:

ID	Descripción	Criticidad
RNF-001	El plug-in deberá ser desarrollado con licencias de código abierto.	Importante
RNF-002	La instalación del plug-in no deberá alterar el rendimiento de la aplicación Redmine.	Importante
RNF-003	El contenido de las consultas en SQL de las métricas no deberá realizar ninguna modificación en la base de datos, es decir, serán únicamente de lectura.	Importante
RNF-004	Las entradas de texto de los formularios de creación y edición de métricas deberán estar protegidos de ataques XSS (Cross-site scripting).	Importante
RNF-005	El plug-in deberá ofrecer un tiempo de respuesta razonable para cualquier operación de consulta.	Importante
RNF-006	El plug-in deberá soportar un mínimo de 15 peticiones simultáneas.	Importante
RNF-007	El plug-in deberá estar disponible en castellano.	Importante

RNF-008	El plug-in deberá estar disponible en inglés.	Importante
---------	---	------------

4.3 Subsistemas y Casos de Uso

4.3.1 Modelo de Subsistemas

A continuación, se muestra un diagrama de los distintos subsistemas en los que se descompone el sistema, con el objetivo de separar para una mejor comprensión de las distintas funcionalidades:

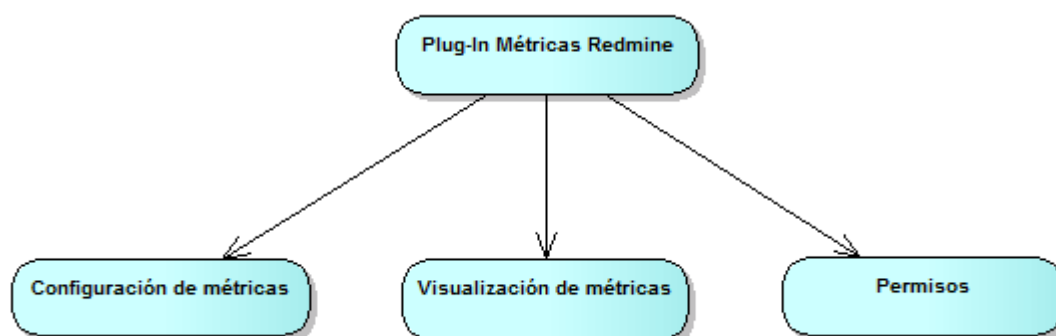


Ilustración 12: Modelo de subsistemas.

Para una mejor comprensión del diagrama, se detalla cada subsistema con las diferentes acciones que se realizarán:

- **Subsistema de configuración de métricas:** Se establecen las distintas opciones que puede realizar el administrador de Redmine y los usuarios con permisos de configuración sobre las métricas. Este subsistema está compuesto de tres funcionalidades:
 - **Funcionalidad de creación de métricas:** Contiene un formulario con los campos necesarios a rellenar. Si el formulario se ha rellenado correctamente se almacenará una nueva métrica en la BBDD.
 - **Funcionalidad de edición de métricas:** Al igual que el subsistema de creación de métricas, contiene un formulario con los campos necesarios a rellenar. Si el formulario se ha rellenado correctamente se actualizará la métrica existente en la BBDD.

- **Funcionalidad de eliminación de métricas:** Ofrece al usuario la opción de eliminar una métrica de la BBDD.
- **Subsistema de visualización:** Conformar la parte de representación de los datos que se obtienen de las métricas establecidas, tanto de forma gráfica como en formato tabla. Se divide en dos subsistemas dependiendo del tipo de usuario que puede acceder a las métricas:
 - **Funcionalidad de visualización de métricas generales:** Añade una nueva pestaña en el menú superior que solo será visible por los usuarios que tengan permiso. En ella se pueden visualizar las métricas generales para todos los proyectos alojados en la aplicación Redmine.
 - **Funcionalidad de visualización de métricas de un proyecto:** Se añade una pestaña en el menú de cada proyecto, en la cual se muestran los datos referentes a las métricas establecidas para ese proyecto específico. A esta pestaña tienen acceso los usuarios que estén asignados a ese proyecto en concreto y posean permisos de visualización.
- **Subsistema de asignación de permisos:** Se añaden las diferentes opciones de permisos que tendrán los usuarios para que el administrador de Redmine pueda asignarle esos permisos a cualquier usuario que él desee. Los permisos se establecen para las funcionalidades de los dos subsistemas anteriores (configuración y visualización de métricas).

4.3.2 Modelo de Casos de Uso

En este punto, se recogerán los casos de uso de los distintos usuarios que pueden tener con el plug-in. Para una mejor comprensión de los casos de uso, se muestran los diagramas correspondientes y a continuación se detalla cada caso de uso en su tabla:

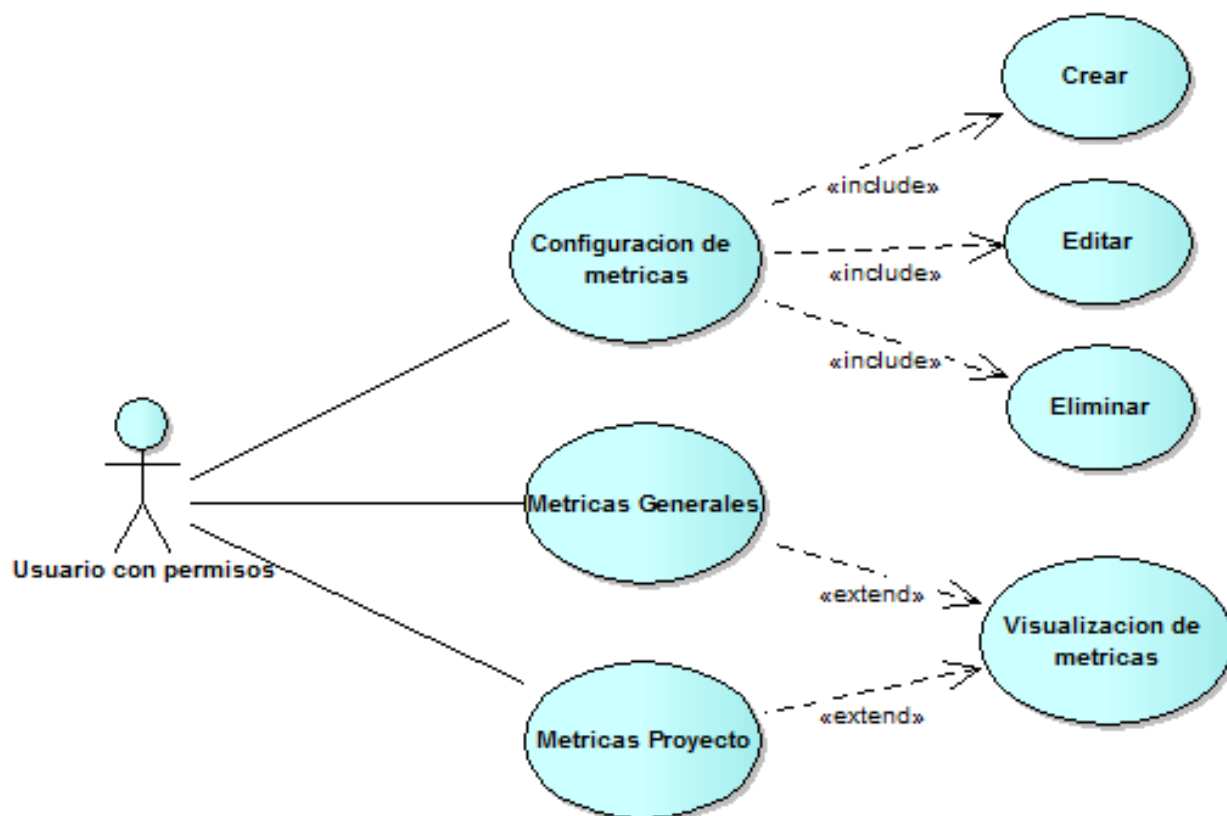


Ilustración 13: Diagrama de casos de uso.

Caso de uso:	Configuración de métricas
Código:	CU-01
Descripción:	<i>Acceso a la configuración de métricas donde se pueden gestionar.</i>
Actores:	
- <i>Usuario con permisos de configuración de métricas.</i>	
Pre-condiciones:	
<i>Debe existir al menos un usuario con permisos de configuración de métricas.</i>	
Flujo Normal:	
<ol style="list-style-type: none"><i>1. El usuario selecciona la opción 'Configuración de métricas' del menú superior.</i><i>2. El usuario escoge el tipo de métricas que desea gestionar.</i><i>3. El sistema muestra un listado de las métricas existentes en la BBDD y sus opciones de editarlas y eliminarlas, así como la opción de crear una nueva.</i>	
Post-condiciones:	
<i>El usuario visualiza las métricas ya creadas y las opciones que se pueden aplicar a las métricas.</i>	

Caso de uso:	Crear nueva métrica
Código:	CU-02
Descripción:	Creación de una nueva métrica en la BBDD.
Actores:	
- Usuario con permiso de creación de métricas	
Pre-condiciones:	
Debe existir al menos un administrador de Redmine o usuario con permiso de creación de métricas.	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario realiza los pasos indicados en el CU-02. 2. El usuario selecciona la opción 'Nueva métrica'. 3. El usuario debe rellenar los campos del formulario correctamente. 4. Una vez rellenado el formulario el usuario debe pulsar el botón 'Crear métrica' para que la métrica se almacene en la BBDD. 5. El sistema mostrará un mensaje informando al usuario si la métrica se ha creado con éxito. 	
Flujo Alternativo:	
Formulario incompleto.	<ol style="list-style-type: none"> 1. Se realizan los pasos 1 y 2 del flujo normal. 2. El usuario no rellena los campos del formulario correctamente. 3. El usuario pulsa el botón 'Crear métrica' para que la métrica se almacene en la BBDD. 4. El sistema muestra un mensaje informando a los usuarios de los campos que no ha rellenado correctamente para que pueda corregirlo.
Error en la creación.	<ol style="list-style-type: none"> 1. Se realizan los pasos 1-4 del flujo normal. 2. El sistema mostrará un mensaje informando al usuario de que la métrica no se ha creado y le redirigirá a la vista de configuración de métricas.
Post-condiciones:	
<p><u>Flujo normal:</u> La nueva métrica creada por el usuario se ha almacenado en la BBDD.</p> <p><u>Formulario incompleto:</u> La métrica no se ha almacenado en la BBDD.</p> <p><u>Error en la creación:</u> La métrica no se ha almacenado en la BBDD.</p>	

Caso de uso:	Editar métrica
Código:	CU-03
Descripción:	<i>Modificación de los parámetros de una métrica ya existente en la BBDD.</i>
Actores:	
- Usuario con permiso de edición de métricas	
Pre-condiciones:	
<i>Debe existir al menos un usuario con permiso de edición de métricas y una métrica creada en la BBDD.</i>	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario realiza los pasos indicados en el CU-02. 2. El usuario selecciona la opción 'Editar' de la métrica que desee modificar. 3. El usuario debe modificar algún dato del formulario correctamente. 4. Una vez rellenado el formulario el usuario debe pulsar el botón 'Actualizar métrica' para que la métrica se actualice en la BBDD. 5. El sistema mostrará un mensaje informando al usuario si la métrica se ha actualizado con éxito. 	
Flujo Alternativo:	
<i>Error en la actualización.</i>	<ol style="list-style-type: none"> 1. Se realizan los pasos 1 y 2 del flujo normal. 2. El usuario debe modificar algún dato del formulario correctamente. 3. El usuario pulsa el botón 'Actualizar métrica' para que la métrica se actualice en la BBDD. 4. El sistema muestra un mensaje informando a los usuarios de los campos que no ha rellenado correctamente para que pueda corregirlo.
<i>Error en la actualización.</i>	<ol style="list-style-type: none"> 1. Se realizan los pasos 1-4 del flujo normal. 2. El sistema mostrará un mensaje informando al usuario de que la métrica no se ha actualizado y le redirigirá a la vista de configuración de métricas.
Post-condiciones:	
<p><u>Flujo normal:</u> La métrica editada por el usuario debe quedar actualizada en la BBDD.</p> <p><u>Error en la actualización:</u> La métrica no se ha actualizado en la BBDD.</p> <p><u>Error en la actualización:</u> La métrica no se ha actualizado en la BBDD.</p>	

Caso de uso:	Borrar métrica
Código:	CU-04
Descripción:	<i>Eliminación de una métrica de la BBDD.</i>
Actores:	
- <i>Usuario con permiso de eliminación de métricas</i>	
Pre-condiciones:	
<i>Debe existir al menos un usuario con permiso de eliminación de métricas y una métrica creada en la BBDD.</i>	
Flujo Normal:	
<ol style="list-style-type: none"> <i>El usuario realiza los pasos indicados en el CU-02.</i> <i>El usuario selecciona la opción 'Borrar' de la métrica que desee eliminar.</i> <i>El sistema pedirá la confirmación del usuario para eliminar la métrica de la BBDD.</i> <i>El sistema mostrará un mensaje informando al usuario si la métrica se ha eliminado con éxito.</i> 	
Flujo Alternativo:	
<i>Error en la eliminación.</i>	<ol style="list-style-type: none"> <i>Se realizan los pasos 1-4 del flujo normal.</i> <i>El sistema mostrará un mensaje informando al usuario de que la métrica no se ha borrado.</i>
Post-condiciones:	
<p><u><i>Flujo normal:</i></u> <i>La métrica se ha eliminado de la BBDD.</i></p> <p><u><i>Error en la eliminación:</i></u> <i>La métrica no se ha eliminado de la BBDD.</i></p>	

Caso de uso:	Visualización de métricas generales
Código:	CU-05
Descripción:	Visualización de los resultados obtenidos de las métricas globales de todos los proyectos.
Actores:	
- Usuario con permiso de visualización de métricas generales	
Pre-condiciones:	
Debe existir al menos un usuario con permiso de visualización de métricas generales y una métrica general creada en la BBDD.	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario accederá a la pestaña 'Métricas generales' situada en el menú superior. 2. El sistema mostrará un listado de todas las métricas generales alojadas en la BBDD. 3. El usuario selecciona una de las métricas. 4. El sistema muestra correctamente los resultados obtenidos por la métrica. 	
Flujo Alternativo:	
Error en la visualización.	<ol style="list-style-type: none"> 1. Se realizan los pasos 1-3 del flujo normal. 2. El sistema muestra un mensaje de error.
Post-condiciones:	
<p><u>Flujo normal:</u> Se muestran los resultados obtenidos en una tabla y un gráfico para esa métrica.</p> <p><u>Error en la visualización:</u> La métrica no ha obtenido resultados que pudiesen ser representados.</p>	

Caso de uso:	Visualización de métricas de un proyecto
Código:	CU-06
Descripción:	Visualización de los resultados obtenidos de las métricas de un proyecto en concreto.
Actores:	
- Usuario con permisos de visualización de métricas de proyecto	
Pre-condiciones:	
<ul style="list-style-type: none"> - Debe existir al menos un usuario. - El usuario estar asignado a ese proyecto y tener permisos de visualización de métricas de proyecto. - Debe existir al menos un proyecto creado en la BBDD. - Debe existir al menos una métrica de proyecto en la BBDD. 	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario accederá a la pestaña 'Métricas' situada en el menú del proyecto. 2. El sistema mostrará un listado de todas las métricas de proyecto alojadas en la BBDD del proyecto en cuestión. 3. El usuario selecciona una de las métricas. 4. El sistema muestra correctamente los resultados obtenidos por la métrica. 	
Flujo Alternativo:	
Error en la visualización.	<ol style="list-style-type: none"> 1. Se realizan los pasos 1-3 del flujo normal. 2. El sistema muestra un mensaje de error.
Post-condiciones:	
<p><u>Flujo normal:</u> Se muestran los resultados obtenidos en una tabla y un gráfico para esa métrica.</p> <p><u>Error en la visualización:</u> La métrica no ha obtenido resultados que pudiesen ser representados.</p>	

4.4 Modelo Lógico de Datos

El Plug-In de métricas sólo deberá almacenar los datos de configuración de cada métrica. Por tanto nuestro modelo de datos del Plug-In se ve reducido a dos entidades: **métricas generales** y **métricas de proyectos**.

La separación de las métricas en dos entidades se debe a que de esta forma podremos gestionar mejor las consultas que contienen, ya que las consultas generales serán diferentes a las de proyecto.

El contenido de nuestras tablas será idéntico, ya que ambas almacenan los datos que se introducen al crear una métrica: título, consulta en SQL, tipo de gráfico, etiquetas del gráfico y valor del gráfico. A continuación se describe cada uno de los atributos de la tabla, para una mejor comprensión:

- **ID:** Este campo se crea automáticamente por el uso del *framework* de Ruby on Rails, e identifica unívocamente cada entrada de la tabla por un número. Este atributo es obligatorio.
- **Título:** Nombre de la métrica que se almacena en forma de cadena de texto.
- **Consulta:** Texto que recoge la consulta SQL que se empleará para obtener los datos.
- **Gráfico:** Será un seleccionable de los diferentes tipos de gráficos que se permitan.
- **Cantidad:** Identifica el tipo de representación que se va a emplear en el gráfico, ya sea simple (una sola serie) o multiserie.
- **Serie:** Indica la columna de datos que va a suponer la serie a representar en los gráficos. Cabe destacar que este atributo solo está disponible en gráficos de representación multiserie en los cuales se dispone de más de un valor para una serie.
- **Categoría:** Recoge en forma de cadena de texto la columna de datos que va a suponer la etiqueta del eje horizontal de los gráficos.
- **Valor:** Cadena de texto que indica la columna de datos que se van a representar y que supone la etiqueta del eje vertical.

4.5 Interfaz de Usuario

A continuación, se identificarán los elementos de la actual interfaz de usuario de Redmine que se verán modificados con la inclusión del Plug-In. También, se describirán los efectos que producen las acciones sobre estos elementos y un *mockup* sencillo.

4.5.1 Menús

Con la inclusión del Plug-In se ven afectados varios menús de la aplicación Redmine. Estos menús sirven para que el usuario pueda navegar para poder ver los resultados de las métricas o para poder configurarlas. En los siguientes apartados se detallan cada uno de los menús involucrados.

4.5.1.1 Menú superior

Se trata del menú principal de la aplicación de Redmine. Con tiene el botón de inicio, una pestaña que recoge todos los proyectos y una opción de ayuda entre otras alternativas. En el caso de que el usuario que acceda a la aplicación posea permisos de administración de Redmine, se ofrecerá visible una pestaña 'Administración', la cual nos permite acceder al menú de administración del que hablaremos en el apartado 7.1.3.

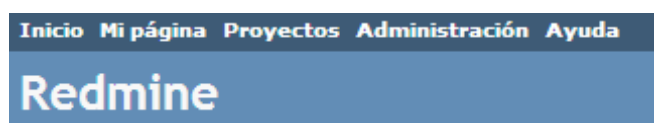


Ilustración 14: Menú superior de Redmine.

El cambio que se producirá en este menú, consistirá en agregar dos nuevas pestañas llamadas 'Métricas generales' y 'Configuración de métricas' que se situarán entre las pestañas 'Proyectos' y 'Administración'. Estas pestañas solo serán accesibles por los usuarios a los que se les haya asignado el permiso de visualización de métricas generales. Al acceder a la pestaña 'Métricas generales', el sistema mostrará un listado de las métricas generales que existen en la BBDD. Por otro lado, la pestaña 'Configuración de métricas' ofrecerá la posibilidad de configurar tanto métricas generales como métricas de proyecto.

A continuación, se muestra un mockup de cómo quedaría el menú superior con la inclusión de esta nueva pestaña:

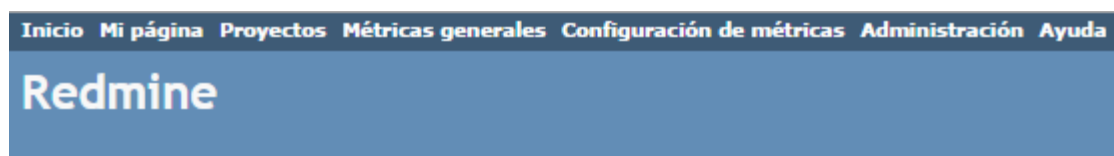


Ilustración 15: Menú superior de Redmine modificado.

4.5.1.2 Menú de proyecto

Este menú recoge todas las opciones que se pueden aplicar sobre un proyecto en Redmine. Como podemos ver en la siguiente imagen, se pueden añadir peticiones, visualizar las ya existentes, observar un diagrama de Gantt con todas las tareas del proyecto, etc.

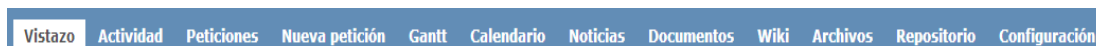


Ilustración 16: Menú de proyecto de Redmine.

En este menú se añadirá una nueva pestaña llamada 'Métricas' que se situará entre las pestañas 'Gantt' y 'Calendario'. Al acceder a esta pestaña, el sistema mostrará un listado de las métricas que existen para ese proyecto. El *mockup* del menú de proyecto es el siguiente:

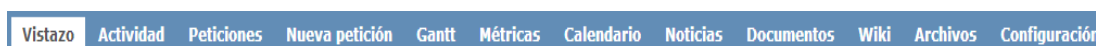


Ilustración 17: Menú de proyecto de Redmine modificado.

4.5.1.3 Configuración de métricas

Una vez seleccionada la opción de 'Configuración de métricas' en el menú superior, se ofrece al usuario la posibilidad de gestionar las métricas generales o las métricas de proyecto:

Configuración de métricas

- Métricas generales
- Métricas de proyecto

Ilustración 18: Interfaz de selección de tipo de métricas.

Al escoger el tipo de métrica que se quiere gestionar, llegamos al apartado de su configuración. Esta parte de la interfaz es totalmente nueva y será necesario crear una vista nueva. En ella se expone un listado con las métricas existentes en la BBDD. Cada una de las métricas irá acompañada de sus acciones de edición y borrado. Además se incluirá una opción para poder crear nuevas métricas que se añadirán a la BBDD. El *mockup* de esta interfaz es el siguiente:

Configuración de métricas

[+ Nueva métrica](#)

Filtro
Título:

Métrica	Opciones
Metrica de prueba	✎ Editar 🗑 Borrار
Segunda metrica de prueba	✎ Editar 🗑 Borrار
Metrica de prueba 3	✎ Editar 🗑 Borrار
Prueba de columnas	✎ Editar 🗑 Borrار
Prueba de columnas 2	✎ Editar 🗑 Borrار
Prueba de tartas	✎ Editar 🗑 Borrار

Ilustración 19: Interfaz de configuración de métricas.

En los siguientes apartados se detallarán cada una de las interfaces que ofrecen las opciones de la configuración de métricas. Cabe destacar que la acción de eliminación de métricas no tiene interfaz y su acción se realiza desde esta interfaz pulsando sobre la opción 'Borrar'.

4.5.1.4 Añadir métrica

Esta interfaz presenta un formulario, en el cual se deben rellenar los campos correctamente para poder crear una nueva métrica.

Nueva métrica

(*) Campos obligatorios

Título *

Consulta SQL *

Representación * ☐ Simple ☐ Multiserie

Ilustración 20: Interfaz de creación de métricas.

Como se puede apreciar en la anterior captura, tenemos dos campos de texto en los cuales podremos introducir el título de la métrica y la consulta SQL que debemos realizar para obtener los datos de la BBDD.

Por otro lado tenemos la posibilidad de seleccionar el tipo de representación del gráfico, ya sea simple o multiserie. Dependiendo del tipo de representación seleccionado se desplegará una serie de campos u otro. Para el tipo simple tenemos la siguiente interfaz:

Nueva métrica

(*) Campos obligatorios



Título *

Consulta SQL *

Representación * ☒ Simple ☐ Multiserie

Atributos del gráfico simple

Tipo de gráfico * ☐ Columnas ☐ Barras ☐ Líneas ☐ Área ☐ Tarta

Categoría *

Valor *

Ilustración 21: Interfaz de creación de métricas de representación simple.

Como se puede apreciar en la captura anterior, al seleccionar la opción de representación simple, se despliega un recuadro que contiene tres campos adicionales: tipo de gráfico, categoría y valor. Cabe destacar, que en el caso de la selección del tipo de gráfico para representaciones simples se admiten todos los tipos de graficos disponibles por chartkick, mientras que para las representaciones multiserie no se encuentra disponible la representación en formato tarta. Esto se puede observar en la interfaz para el tipo tipo multiserie que se muestra a continuación:

Nueva métrica

(*) Campos obligatorios

Título *

Consulta SQL *

Representación *

☐ Simple ☒ Multiserie

Atributos del gráfico multiserie

Tipo de gráfico *

☐ Columnas ☐ Barras ☐ Líneas ☐ Área

Categoría *

Selecione ▼

Serie *

Selecione ▼

Valor *

Selecione ▼

Crear métrica

Ilustración 22: Interfaz de creación de métricas de representación multiserie.

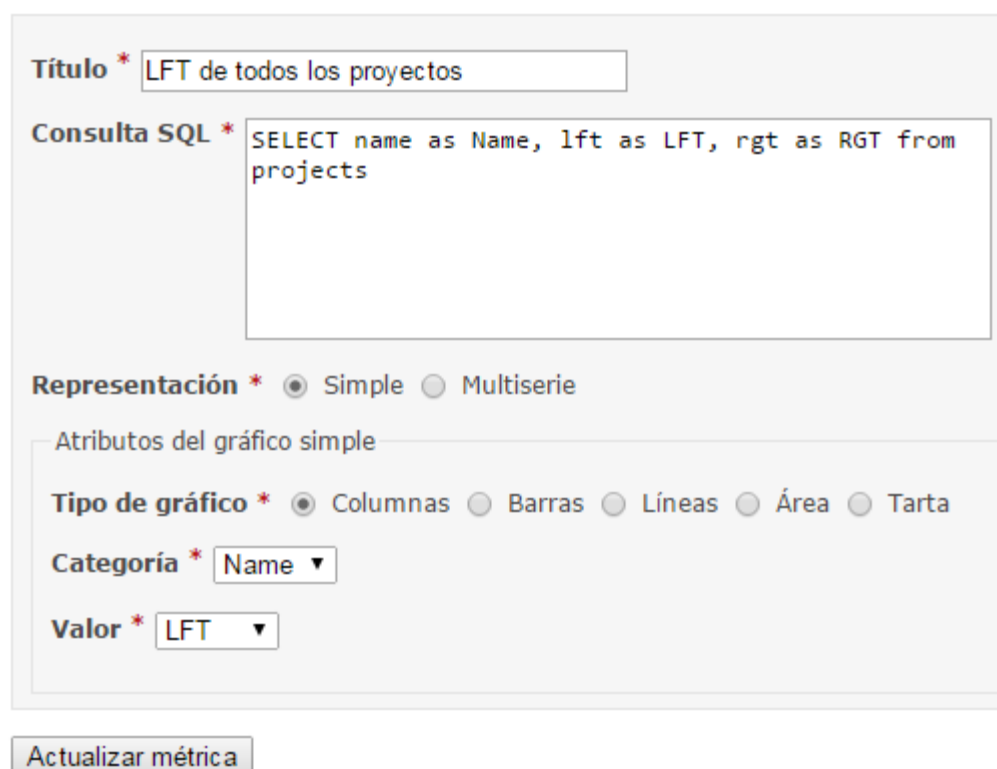
En el caso de la representación multiserie, nos encontramos con cuatro campos a rellenar por el usuario: tipo de gráfico, categoría, serie y valor. A diferencia del caso de representación simple, en la representación multiserie tenemos el campo serie, el cual determina la columna de datos que se quiere representar para los valores asociados.

4.5.1.5 Editar métrica

La interfaz de edición de métricas es prácticamente idéntica a la de creación de métricas. Las diferencias las encontramos en que los campos a rellenar en el formulario se presentan completados con los datos que tenía la métrica almacenada y así facilitar la labor de edición.

Editar métrica

(*) Campos obligatorios



Título * LFT de todos los proyectos

Consulta SQL * `SELECT name as Name, lft as LFT, rgt as RGT from projects`

Representación * ☒ Simple ☐ Multiserie

Atributos del gráfico simple

Tipo de gráfico * ☒ Columnas ☐ Barras ☐ Líneas ☐ Área ☐ Tarta

Categoría * Name ▼

Valor * LFT ▼

Actualizar métrica

Ilustración 23: Interfaz de edición de métricas.

4.5.2 Visualización de métricas

Una vez que hemos creado una métrica, se podrá visualizar los datos que se obtienen de ella. El acceso a este tipo de interfaz se encuentra tanto en el menú superior (acceso a métricas generales), como en el menú de proyecto (acceso a métricas de proyecto). La diferencia del acceso tiene que ver con el tipo de usuario, ya que a través del menú superior solo pueden acceder usuarios con permiso para visualizar todas las métricas generales para todos los proyectos; mientras que los usuarios asignados a un proyecto, solo podrán acceder a la visualización de las métricas que existen para su proyecto si tienen permiso para ello. A continuación, se muestran las dos interfaces encargadas de ofrecer el listado de métricas según el tipo:

Métricas generales

Filtro

Título:

Aceptar

 Anular

Métricas
Metrica de prueba
Segunda metrica de prueba
Metrica de prueba 3
Prueba de columnas
Prueba de columnas 2
Prueba de tartas
Prueba de columnas 3

Ilustración 24: Interfaz de visualización de métricas generales.

Métricas de proyecto

Filtro

Título:

Aceptar

Anular

Métricas
Metrica de prueba
Segunda metrica de prueba
Metrica de prueba 3
Prueba de columnas
Prueba de columnas 2
Prueba de tartas
Prueba de columnas 3

Ilustración 25: Interfaz de visualización de métricas de proyecto.

Cada entrada del listado corresponde a una métrica a la cual podremos acceder a ella para visualizar los datos. Como se ha especificado en los requisitos la representación de los datos de las métricas consta de dos partes: la tabla de datos y el gráfico seleccionado. Además, incorporará las opciones de exportación de los datos en formato CSV y PDF. El siguiente *mockup* muestra la visualización de una métrica general:

LFT de todos los proyectos

Gráfica de los datos obtenidos:

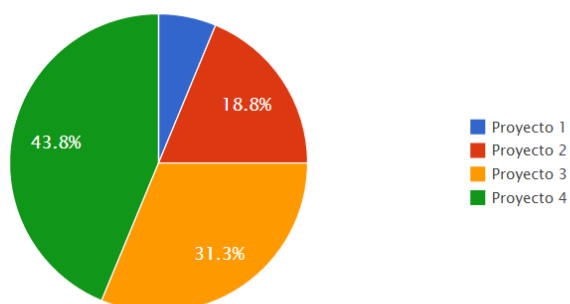


Tabla de resultados:

name	lft	rgt
Proyecto 1	1	2
Proyecto 2	3	4
Proyecto 3	5	6
Proyecto 4	7	8

Exportar a: CSV | PDF

Ilustración 26: Interfaz de datos de una métrica.

4.5.3 Configuración de permisos

La configuración de los permisos de Redmine se realiza a través del menú de administración. Existen dos formas de agregar permisos a un usuario: creando un nuevo perfil o asignándole permisos a uno existente. A continuación, se muestran dos imágenes con los dos casos de asignación de permisos:

Perfiles » Nuevo perfil

Nombre *

Se pueden asignar ☒ peticiones a este perfil

Visibilidad de las peticiones Todas las peticiones no privadas ▼

Copiar flujo de trabajo desde ▼

Permisos

Proyecto

☐ Crear proyecto
 ☐ Modificar proyecto
 ☐ Cerrar / reabrir el proyecto

☐ Seleccionar módulos del proyecto
 ☐ Administrar miembros
 ☐ Administrar versiones

☐ Crear subproyectos
 ☐ Ver métricas
 ☐ Crear métricas

☐ Editar métricas
 ☐ Borrar métricas
 ☐

Foros

☐ Administrar foros
 ☒ Enviar mensajes
 ☐ Modificar mensajes

☐ Editar mensajes propios
 ☐ Borrar mensajes
 ☐ Borrar mensajes propios

Ilustración 27: Interfaz de creación de perfil.

Perfiles » Informe de permisos

Permisos	Jefe de proyecto ✓	Desarrollador ✓	Informador ✓	No miembro ✓
✓ Crear proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Modificar proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Cerrar / reabrir el proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Seleccionar módulos del proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar miembros	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar versiones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Crear subproyectos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 28: Interfaz de asignación de permisos.

Las dos capturas anteriores serán las interfaces que deberemos modificar para que el administrador pueda otorgar los permisos sobre las métricas. Acto seguido, se muestran dos *mockups* de cómo serían estas opciones que hay que añadirle a la interfaz:

Perfiles » Nuevo perfil

Nombre *

Se pueden asignar peticiones a este perfil ☒

Visibilidad de las peticiones

Copiar flujo de trabajo desde

Permisos

Proyecto

<input type="checkbox"/> Crear proyecto	<input type="checkbox"/> Modificar proyecto	<input type="checkbox"/> Cerrar / reabrir el proyecto
<input type="checkbox"/> Seleccionar módulos del proyecto	<input type="checkbox"/> Administrar miembros	<input type="checkbox"/> Administrar versiones
<input type="checkbox"/> Crear subproyectos	<input checked="" type="checkbox"/> Visualizar métricas generales	<input checked="" type="checkbox"/> Configurar métricas generales
<input checked="" type="checkbox"/> Configurar métricas de proyecto		

Ilustración 29: Componente de la interfaz de creación de perfil.

Perfiles » Informe de permisos

Permisos	Jefe de proyecto ✓	Desarrollador ✓	Informador ✓
✓ Crear proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Modificar proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Cerrar / reabrir el proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Seleccionar módulos del proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar miembros	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar versiones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Crear subproyectos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Visualizar métricas generales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Configurar métricas generales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Configurar métricas de proyecto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 30: Componente de la interfaz de asignación de permisos.

Mettricas	Jefe de proyecto	Desarrollador	Informador
✓ Visualizar métricas de proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 31: Permisos del módulo Métricas.

4.5.4 Matrices de trazabilidad

En los siguientes apartados se presentan las matrices de trazabilidad requeridas.

4.5.4.1 Requisitos

ID	Origen	Descripción	Dependencias con otros Requisitos
RU-001	Análisis	El usuario debe conocer el lenguaje SQL para poder crear métricas.	RF-005
RF-001	Análisis	El usuario deberá tener los permisos necesarios para poder realizar las acciones que ofrece el plug-in.	RF-002, RF-003, RF-004, RF-012 ,RF-013, RF-017, RF-018, RF-019, RF-020, RF-021
RF-002	Análisis	Un usuario podrá crear métricas nuevas si tiene permiso.	RF-001, RF-005
RF-003	Análisis	Un usuario podrá editar métricas ya existentes si tiene permiso.	RF-001
RF-004	Análisis	Un usuario podrá eliminar métricas si tiene permiso.	RF-001
RF-005	Análisis	El usuario deberá rellenar un formulario para crear una métrica.	RF-002, RF-006, RF-007, RF-008, RNF-003
RF-006	Análisis	El sistema deberá mostrar un mensaje de error si falta algún campo obligatorio por rellenar en el formulario de métricas.	RF-005
RF-007	Análisis	El sistema deberá mostrar un mensaje de error si algún campo se ha rellenado de forma incorrecta en el formulario de métricas.	RF-005

RF-008	Análisis	El sistema deberá mostrar un mensaje de que la métrica ha sido creada con éxito.	RF-005
RF-009	Análisis	El sistema deberá mostrar un mensaje de que la métrica ha sido editada con éxito.	RF-003
RF-010	Análisis	El sistema deberá mostrar un mensaje de confirmación de eliminación de la métrica.	RF-004
RF-011	Análisis	El sistema deberá mostrar un mensaje de que la métrica ha sido eliminada con éxito.	RF-010
RF-012	Análisis	Un usuario asociado a un proyecto podrá ver las métricas asociadas a dicho proyecto si tiene permisos para ello.	RF-001
RF-013	Análisis	Un usuario podrá visualizar métricas generales de todos los proyectos de Redmine si tiene permisos para ello.	RF-001
RF-014	Análisis	El sistema deberá mostrar en la parte de visualización de métricas los datos obtenidos por la consulta de una métrica en una tabla.	RF-008
RF-015	Análisis	El sistema deberá mostrar en la parte de visualización de métricas los datos obtenidos por la consulta de una métrica en una gráfica, cuyo tipo fue indicado en la creación de la métrica.	RF-008

RF-016	Análisis	El sistema deberá mostrar un mensaje de error al usuario si se produce un error al ejecutar la consulta de la métrica.	RF-014, RF-015
RF-017	Análisis	El sistema deberá mostrar una nueva opción de menú “Configuración de métricas” dentro del menú “Administración”, la cual solo será visible por los usuarios que tengan permisos asignados.	RF-001
RF-018	Análisis	El sistema deberá mostrar la opción de crear una nueva métrica dentro del menú “Configuración de métricas” que será visible por los usuarios que tengan permisos asignados.	RF-001, RF-017
RF-019	Análisis	El sistema deberá mostrar la opción de editar una métrica dentro del menú “Configuración de métricas”, que será visible por los usuarios que tengan permisos asignados.	RF-001, RF-017
RF-020	Análisis	El sistema deberá mostrar la opción de eliminar una métrica dentro del menú “Configuración de métricas”, que será visible por los usuarios que tengan permisos asignados.	RF-001, RF-017

RF-021	Análisis	El sistema deberá mostrar en el menú superior una nueva pestaña llamada “Métricas generales”, que será visible por los usuarios que tengan permisos asignados.	RF-001
RF-022	Análisis	El sistema deberá mostrar un listado de las métricas generales almacenadas en la base datos en la pestaña “Métricas generales” del menú superior.	RF-021
RF-023	Análisis	El sistema deberá mostrar en el menú de proyecto una nueva pestaña llamada “Métricas”, que solo será visible por los usuarios que tengan acceso a dicho proyecto y tengan permisos de visualización.	RF-001
RF-024	Análisis	El sistema deberá mostrar un listado de las métricas del proyecto al que se ha accedido a través de la pestaña “Métricas” del menú del proyecto.	RF-023
RF-025	Análisis	El sistema ofrecerá al usuario la opción de poder exportar los resultados obtenidos en forma tabular de la métrica que puede visualizar en un fichero con formato CSV.	RF-014, RF-015

RF-026	Análisis	El sistema ofrecerá al usuario la opción de poder exportar los resultados obtenidos de la métrica que puede visualizar en un fichero con formato PDF.	RF-014, RF-015
RNF-001	Análisis	El plug-in deberá ser desarrollado con licencias de código abierto.	
RNF-002	Análisis	La instalación del plug-in no deberá alterar el rendimiento de la aplicación Redmine.	
RNF-003	Análisis	El contenido de las consultas en SQL de las métricas no deberá realizar ninguna modificación en la base de datos, es decir, serán únicamente de lectura.	RF-005
RNF-004	Análisis	Las entradas de texto de los formularios de creación y edición de métricas deberán estar protegidos de ataques XSS (Cross-site scripting).	RF-005
RNF-005	Análisis	El plug-in deberá ofrecer un tiempo de respuesta razonable para cualquier operación de consulta.	RF-014, RF-015
RNF-006	Análisis	El plug-in deberá soportar un mínimo de 15 peticiones simultáneas.	
RNF-007	Análisis	El plug-in deberá estar disponible en castellano.	
RNF-008	Análisis	El plug-in deberá estar disponible en inglés.	

4.5.4.2 Requisitos – Casos de uso

ID Requisito	Caso de Uso					
	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06
RU-001		X	X			
RF-001	X	X	X	X	X	X
RF-002		X				
RF-003			X			
RF-004				X		
RF-005		X				
RF-006		X	X			
RF-007		X	X			
RF-008		X				
RF-009			X			
RF-010				X		
RF-011				X		
RF-012						X
RF-013					X	
RF-014					X	X
RF-015					X	X
RF-016					X	X
RF-017	X					
RF-018	X					
RF-019	X					
RF-020	X					
RF-021					X	
RF-022					X	
RF-023						X
RF-024						X
RF-025					X	X
RF-026					X	X
RNF-001						
RNF-002						
RNF-003		X	X			
RNF-004		X	X			
RNF-005	X				X	X
RNF-006						
RNF-007	X	X	X	X	X	X

RNF-008	X	X	X	X	X	X
---------	---	---	---	---	---	---

4.5.4.3 Requisitos – Módulos

ID Requisito	Módulo del Sistema		
	Configuración de métricas	Visualización de métricas	Asignación de permisos
RU-001	X		
RF-001			X
RF-002	X		X
RF-003	X		X
RF-004	X		X
RF-005	X		
RF-006	X		
RF-007	X		
RF-008	X		
RF-009	X		
RF-010	X		
RF-011	X		
RF-012		X	X
RF-013		X	X
RF-014		X	
RF-015		X	
RF-016		X	
RF-017	X		X
RF-018	X		X
RF-019	X		X
RF-020	X		X
RF-021		X	X
RF-022		X	
RF-023		X	X
RF-024		X	
RF-025		X	
RF-026		X	
RNF-001	X	X	X
RNF-002	X	X	X
RNF-003	X	X	



RNF-004	X		
RNF-005	X	X	X
RNF-006	X	X	X
RNF-007	X	X	X
RNF-008	X	X	X

4.5.4.4 Requisitos – Pruebas

4.5.4.4.1 Pruebas 1 - 10

ID Requisito	Pruebas									
	PU-001	PU-002	PU-003	PU-004	PU-005	PU-006	PU-007	PU-008	PU-009	PU-010
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										
RF-005	X	X	X	X	X	X	X	X	X	X
RF-006	X	X	X	X	X	X	X	X	X	X
RF-007	X	X	X	X	X	X	X	X	X	X
RF-008	X	X	X	X	X	X	X	X	X	X
RF-009	X	X	X	X	X	X	X	X	X	X
RF-010										
RF-011										
RF-012										
RF-013										
RF-014										
RF-015										
RF-016										
RF-017										
RF-018										
RF-019										
RF-020										

RF-021										
RF-022										
RF-023										
RF-024										
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003					X	X	X	X	X	X
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.2 Pruebas 11 - 20

ID Requisito	Pruebas									
	PU-011	PU-012	PU-013	PU-014	PU-015	PU-016	PU-017	PU-018	PU-019	PU-020
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										
RF-005	X	X	X	X	X	X	X	X	X	X
RF-006	X	X	X	X	X	X	X	X	X	X



RF-007	X	X	X	X	X	X	X	X	X	X
RF-008	X	X	X	X	X	X	X	X	X	X
RF-009	X	X	X	X	X	X	X	X	X	X
RF-010										
RF-011										
RF-012										
RF-013										
RF-014										
RF-015										
RF-016										
RF-017										
RF-018										
RF-019										
RF-020										
RF-021										
RF-022										
RF-023										
RF-024										
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003	X	X	X	X	X	X	X	X	X	X
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.3 Pruebas 21 - 30

ID Requisito	Pruebas									
	PU-021	PU-022	PU-023	PU-024	PU-025	PU-026	PU-027	PU-028	PU-029	PU-030
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										
RF-005	X	X	X	X	X	X	X	X	X	X
RF-006	X	X	X	X	X	X	X	X	X	X
RF-007	X	X	X	X	X	X	X	X	X	X
RF-008	X	X	X	X	X	X	X	X	X	X
RF-009	X	X	X	X	X	X	X	X	X	X
RF-010										
RF-011										
RF-012										
RF-013										
RF-014										
RF-015										
RF-016										
RF-017										
RF-018										
RF-019										
RF-020										
RF-021										
RF-022										

RF-023										
RF-024										
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003	X	X	X	X	X					
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.4 Pruebas 31 - 40

ID Requisito	Pruebas									
	PU-031	PU-032	PU-033	PU-034	PU-035	PU-036	PU-037	PU-038	PU-039	PU-040
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										
RF-005	X	X	X	X	X	X				
RF-006	X	X	X	X	X	X				
RF-007	X	X	X	X	X	X				
RF-008	X	X	X	X	X	X				



RF-009	X	X	X	X	X	X				
RF-010										
RF-011										
RF-012										
RF-013										
RF-014										
RF-015										
RF-016										
RF-017										
RF-018										
RF-019										
RF-020										
RF-021										
RF-022							X	X	X	X
RF-023										
RF-024										
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003										
RNF-004			X	X	X	X				
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.5 Pruebas 41 - 50

ID Requisito	Pruebas									
	PU-041	PU-042	PU-043	PU-044	PU-045	PU-046	PU-047	PU-048	PU-049	PU-050
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										X
RF-005										
RF-006										
RF-007										
RF-008										
RF-009										
RF-010										
RF-011										
RF-012										
RF-013										
RF-014					X					
RF-015						X	X			
RF-016								X		
RF-017										
RF-018									X	
RF-019										
RF-020										
RF-021										
RF-022										



RF-023										
RF-024	X	X	X	X						
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003										
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.6 Pruebas 51 - 60

ID Requisito	Pruebas									
	PU-051	PU-052	PU-053	PU-054	PU-055	PU-056	PU-057	PU-058	PU-059	PU-060
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										
RF-005										
RF-006										
RF-007										
RF-008										



RF-009		X								
RF-010			X							
RF-011				X						
RF-012										
RF-013										
RF-014										
RF-015										
RF-016										
RF-017										
RF-018										
RF-019	X									
RF-020										
RF-021										
RF-022					X	X	X			
RF-023										
RF-024								X	X	X
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003										
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.7 Pruebas 61 - 70

ID Requisito	Pruebas									
	PU-061	PU-062	PU-063	PU-064	PU-065	PU-066	PU-067	PU-068	PU-069	PU-070
RU-001										
RF-001										
RF-002										
RF-003										
RF-004										
RF-005										
RF-006										
RF-007										
RF-008										
RF-009										
RF-010										
RF-011										
RF-012					X	X				
RF-013			X	X						
RF-014										
RF-015										
RF-016										
RF-017							X	X	X	X
RF-018										
RF-019										
RF-020										
RF-021	X		X	X						
RF-022										



RF-023		X			X	X				
RF-024										
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003										
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.8 Pruebas 71 - 80

ID Requisito	Pruebas									
	PU-071	PU-072	PU-073	PU-074	PU-075	PU-076	PU-077	PU-078	PU-079	PU-080
RU-001										
RF-001										
RF-002			X	X					X	X
RF-003					X	X				
RF-004										
RF-005										
RF-006										
RF-007										
RF-008										



RF-009										
RF-010										
RF-011										
RF-012										
RF-013										
RF-014										
RF-015										
RF-016										
RF-017	X	X								
RF-018			X	X					X	X
RF-019					X	X				
RF-020							X	X		
RF-021										
RF-022										
RF-023										
RF-024										
RF-025										
RF-026										
RNF-001										
RNF-002										
RNF-003										
RNF-004										
RNF-005										
RNF-006										
RNF-007										
RNF-008										

4.5.4.4.9 Pruebas 81 - 91

ID Requisito	Pruebas										
	PU-081	PU-082	PU-083	PU-084	PU-085	PU-086	PU-087	PU-088	PU-089	PU-090	PU-091
RU-001											
RF-001					X						
RF-002											
RF-003	X	X									
RF-004											
RF-005											
RF-006											
RF-007											
RF-008											
RF-009											
RF-010											
RF-011											
RF-012											
RF-013											
RF-014											
RF-015											
RF-016											
RF-017											
RF-018											
RF-019	X	X									
RF-020			X	X							
RF-021											
RF-022											



RF-023											
RF-024											
RF-025						X	X				
RF-026								X	X		
RNF-001											
RNF-002											
RNF-003											
RNF-004											
RNF-005											
RNF-006											
RNF-007										X	
RNF-008											X

5 | Diseño

5.1 Introducción

El presente documento tiene como objetivo definir la arquitectura del sistema y los planteamientos lógicos que se han dado durante la fase de análisis del proyecto “**Métricas Redmine**”. Se va a esclarecer cómo será el diseño del Plug-In que queremos desarrollar.

Se definirá cómo solucionar los problemas planteados, detallando todo lo necesario para la buena comprensión del mismo, como pueden ser los algoritmos o las técnicas y procedimientos concretos que usemos.

Una gran parte del mismo, conllevará la creación y explicación de los componentes del sistema, pues es una la base de la aplicación y deben estar lo suficientemente claros para poder realizar un buen diseño y construcción del sistema.

Una vez se tiene la suficiente información recogida sobre las bases de nuestro sistema, se generarán todas las especificaciones detalladas de la construcción relativa al sistema.

5.2 Definición de la Arquitectura del Sistema

5.2.1 Definición de la Arquitectura Software

La arquitectura que seguirá nuestro sistema será **Modelo-Vista-Controlador** (MVC). Hay que destacar que tanto el Plug-In de métricas que vamos a desarrollar, como la propia aplicación Redmine, emplean el *framework* **Ruby on Rails**, el cual emplea dicha arquitectura.

Con esta arquitectura de software separaremos nuestra capa de negocio (controlador), de la capa de presentación (vista) y de la capa de datos (modelo).

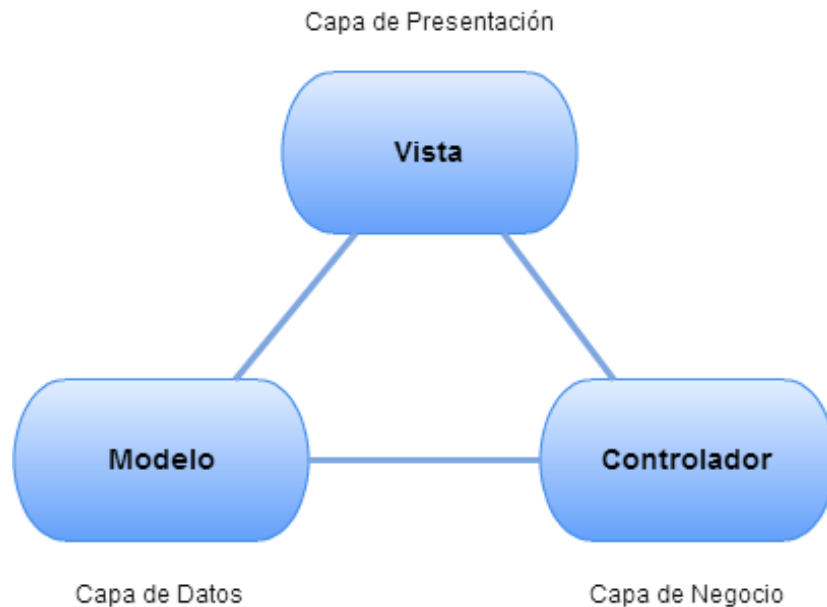


Ilustración 32: Arquitectura MVC.

A continuación, se detallarán cada uno de los tres componentes de la arquitectura MVC de nuestro Plug-In:

- **Modelo:** Es la representación de la información con la cual el sistema opera, más concretamente en las clases que representan a las tablas de la base de datos. Gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). En Ruby on Rails, las clases del Modelo son gestionadas por **ActiveRecord** (heredan de la *clase ActiveRecord::Base*). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **Vista:** En MVC es la lógica de visualización. Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para que el usuario pueda interactuar con ella (usualmente con la interfaz de usuario). En Ruby on Rails las vistas se muestran en archivos con extensión *“.html.erb”*, los cuales están compuesto de código HTML con código de Ruby empotrado.
- **Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un registro en una base de datos). La implementación del 'controlador' es manejada por el **ActionPack** de Rails, que contiene la clase **ApplicationController**. Una aplicación Rails simplemente hereda de esta clase y define las acciones necesarias como métodos, que pueden ser invocados desde la web, por lo general en

la forma `http://aplicacion/ejemplo/metodo`, que invoca a `EjemploController#método`, y presenta los datos usando el archivo de plantilla `/app/views/ejemplo/método.html.erb`.

5.2.1.1 Diagrama de subsistemas

En este apartado se mostrarán los diagramas de subsistemas, de acuerdo a los subsistemas especificados en el apartado 5.1 “Modelo de subsistemas” del documento de análisis.

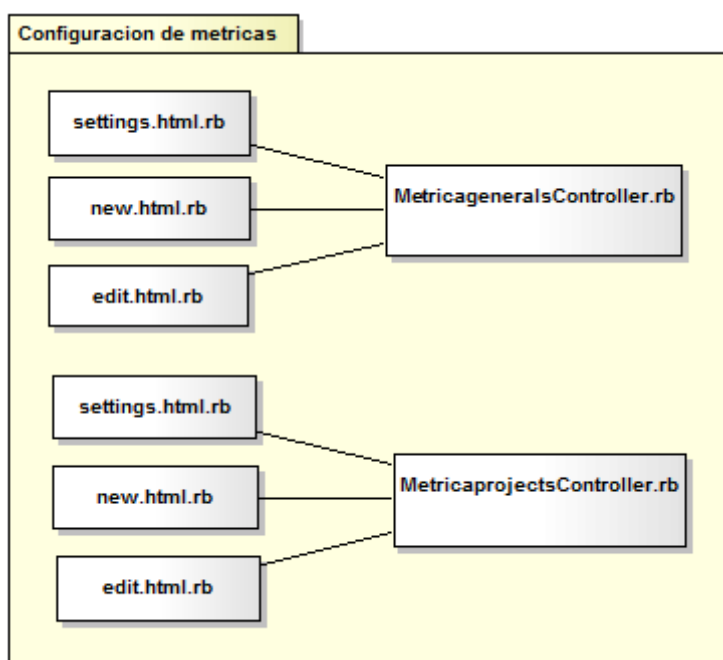


Ilustración 33: Subsistema de configuración de métricas.

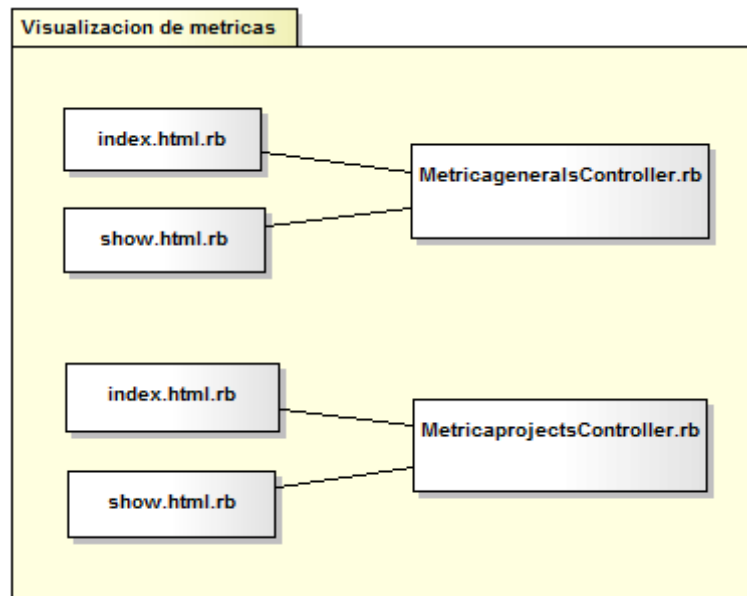


Ilustración 34: Subsistema de visualización de métricas.

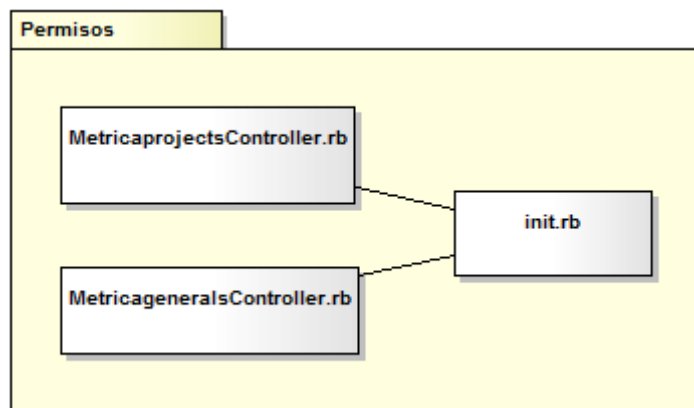


Ilustración 35: Subsistema de permisos.

A continuación, se detalla la composición de cada uno de los subsistemas:

- **Configuración de métricas:** Este subsistema tiene como objetivo ofrecer a los usuarios con permisos de configuración de métricas, las opciones de crear, editar y borrar métricas en la BBDD de Redmine. Este subsistema incluye tres vistas por cada controlador, que recogen las opciones de configuración de métricas. Estas vistas se comunican con su respectivo controlador, ya sean de métricas generales (`MetricageneralsController.rb`) o de proyecto (`MetricaprojectsController.rb`). Dichos controladores contienen los servicios necesarios para llevar a cabo las labores de configuración. Por último el controlador se comunica con el modelo de datos correspondiente para realizar los cambios oportunos en la BBDD.

- **Visualización de métricas:** En este subsistema se recoge la funcionalidad para poder visualizar cada una de las métricas. Se incluyen dos vistas por cada controlador, de las cuales las vistas “index” muestran un listado de las métricas generales y de cada proyecto respectivamente; mientras que las vistas “show” muestran el contenido de la métrica, es decir, los datos obtenidos por ella de forma tabular y gráfica. Las vistas se comunican con los controladores “MetricageneralsController.rb” y “MetricageneralsController.rb” para solicitar sus servicios que permiten gestionar el acceso al modelo de datos.
- **Permisos:** Se establecen los permisos correspondientes sobre las labores que se realizan en los dos subsistemas citados anteriormente. Los módulos de código, que contienen la asignación de permisos, se deberán añadir en la clase “init.rb” la cual contiene la configuración inicial del plug-in. Los permisos indicados en la clase “init.rb” actúan directamente con los métodos que forman parte de su controlador correspondiente.

Para conocer mejor como funciona cada uno de los subsistemas del plug-in, se muestran los diagramas de secuencia correspondientes.

5.2.1.2 Diagramas de secuencia

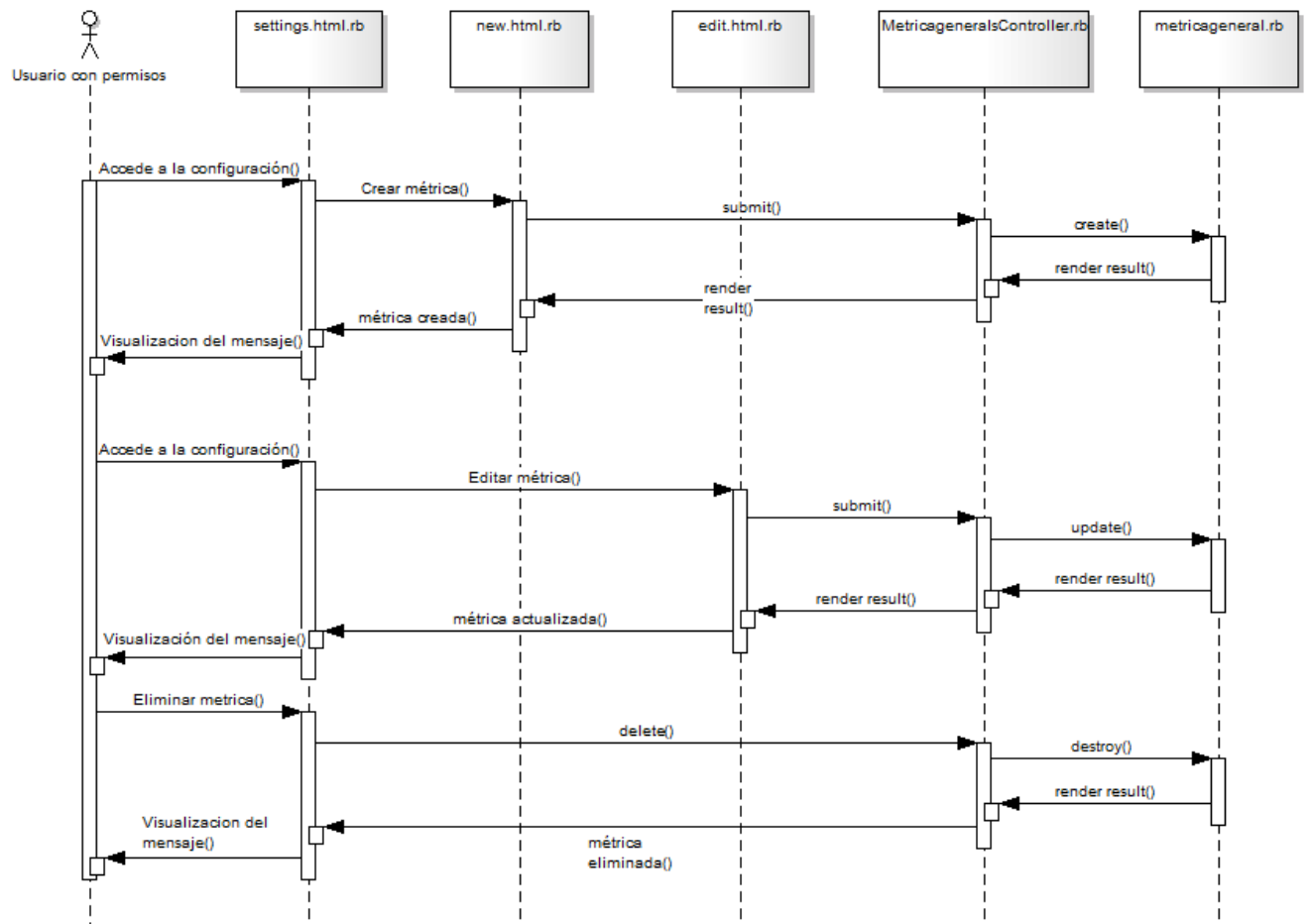


Ilustración 36: Diagrama de secuencia de configuración de métricas generales.

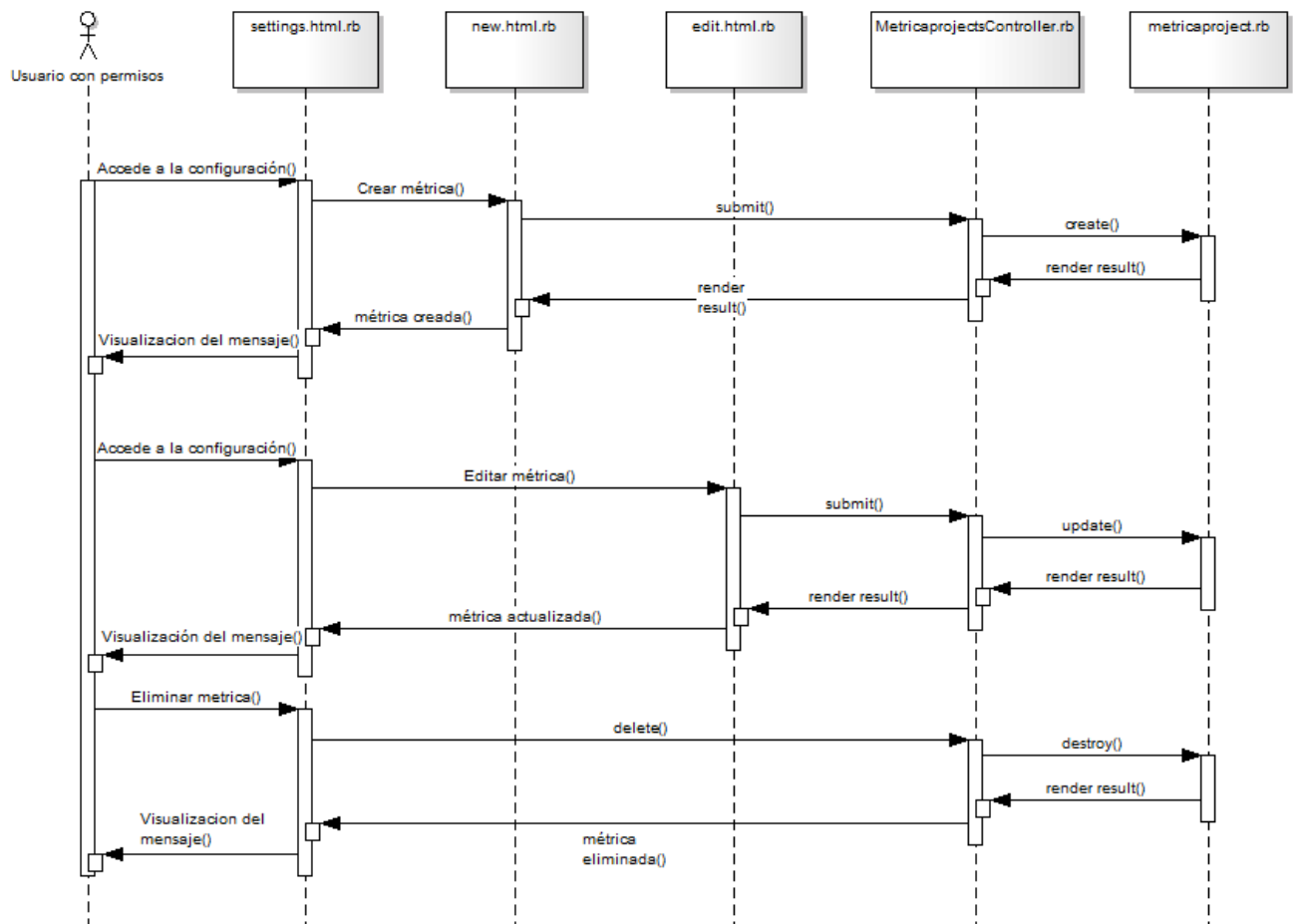


Ilustración 37: Diagrama de secuencia de configuración de métricas de proyecto.

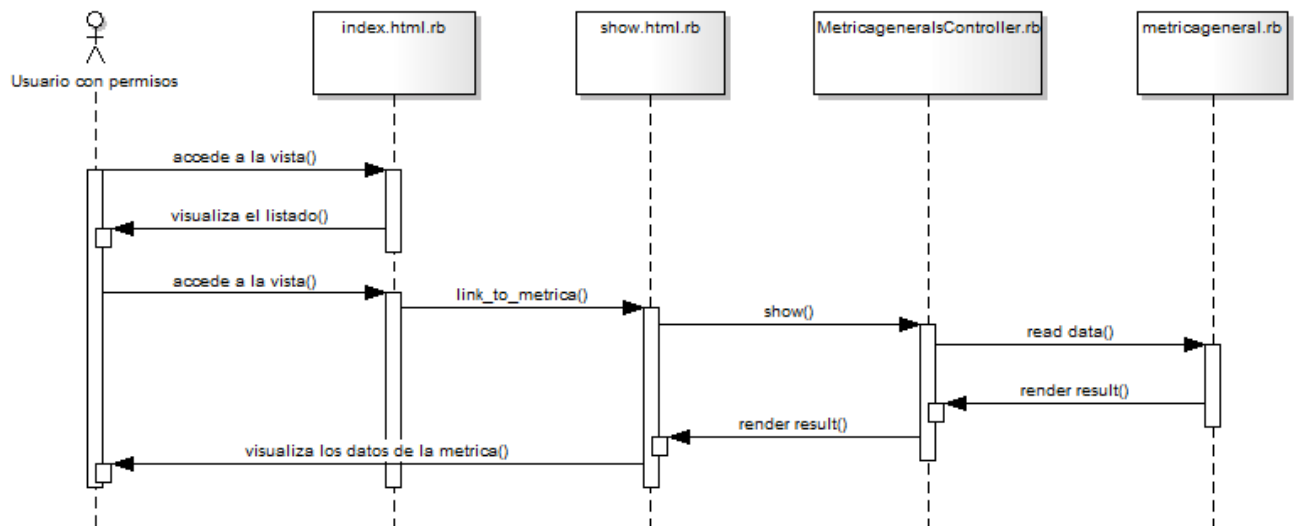


Ilustración 38: Diagrama de secuencia de visualización de métricas generales.

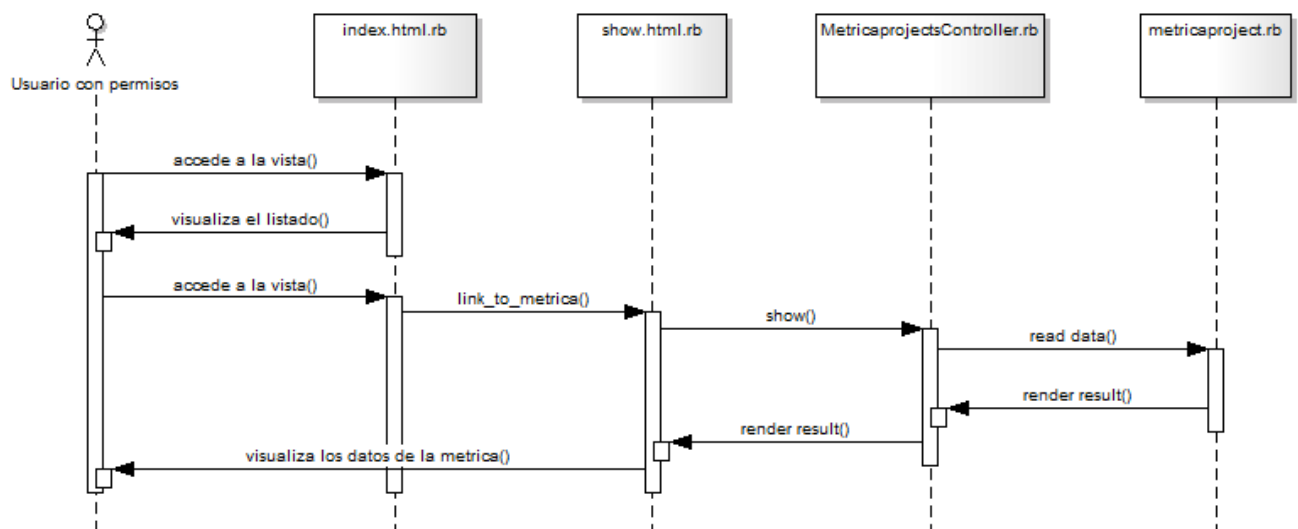


Ilustración 39: Diagrama de secuencia de visualización de métricas de proyecto.

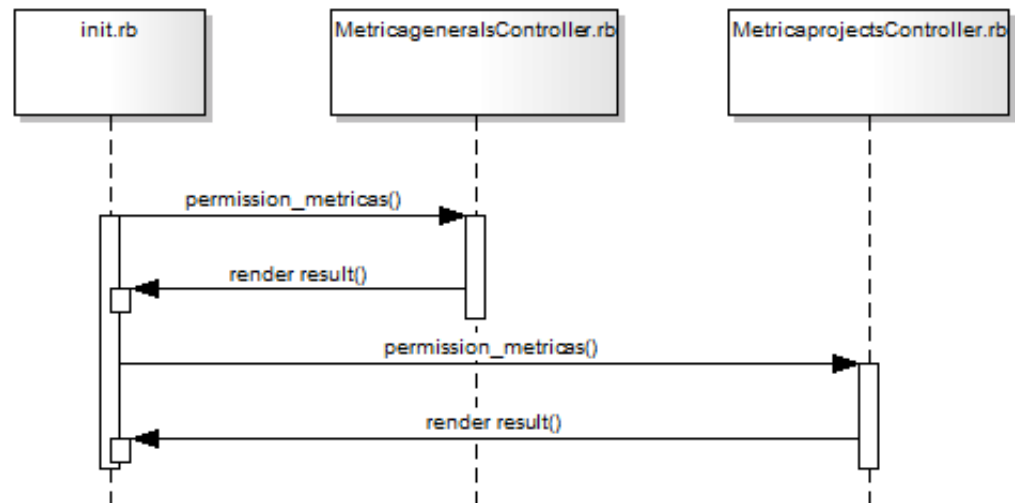


Ilustración 40: Diagrama de secuencia de permisos.

5.3 Diseño de la Capa de Presentación

En los siguientes apartados se procederá a mostrar una explicación más detallada sobre cómo funciona la interfaz de usuario de nuestro plug-in.

5.3.1 Diagramas de Navegación

En este apartado se mostrarán los diagramas correspondientes a la navegación entre los distintos elementos de la interfaz de usuario. Para una mejor organización, se dividirán los diagramas según el subsistema al que pertenecen.

5.3.1.1 Subsistema de configuración de métricas

A continuación se muestra el diagrama de navegación que muestra el acceso a la configuración de las métricas:



Ilustración 41: Diagrama de navegación de configuración de métricas.

El siguiente diagrama de navegación muestra la creación de una métrica:

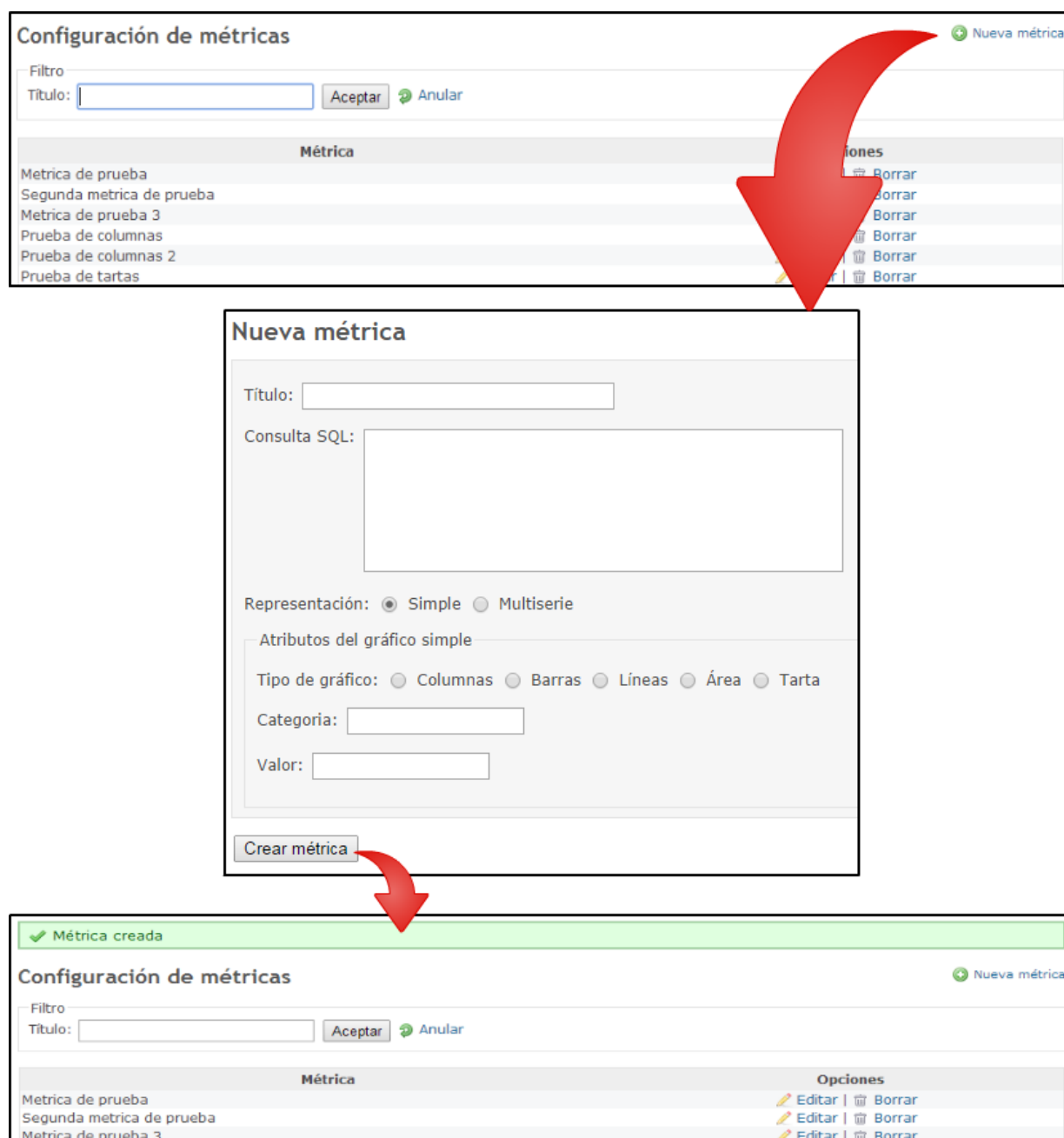


Ilustración 42: Diagrama de navegación de creación de métricas.

En cuanto a la edición de una métrica, el diagrama de navegación es el siguiente:



Ilustración 43: Diagrama de navegación de edición de métricas.

Para la eliminación de métricas, tenemos el siguiente diagrama de navegación:



Ilustración 44: Diagrama de navegación de eliminación de métricas.

5.3.1.2 Subsistema de visualización de métricas

A continuación, se muestra el diagrama de navegación que representa la visualización de métricas generales:



Ilustración 45: Diagrama de navegación de visualización de métricas generales.

Para la visualización de métricas de proyecto tenemos el siguiente diagrama:

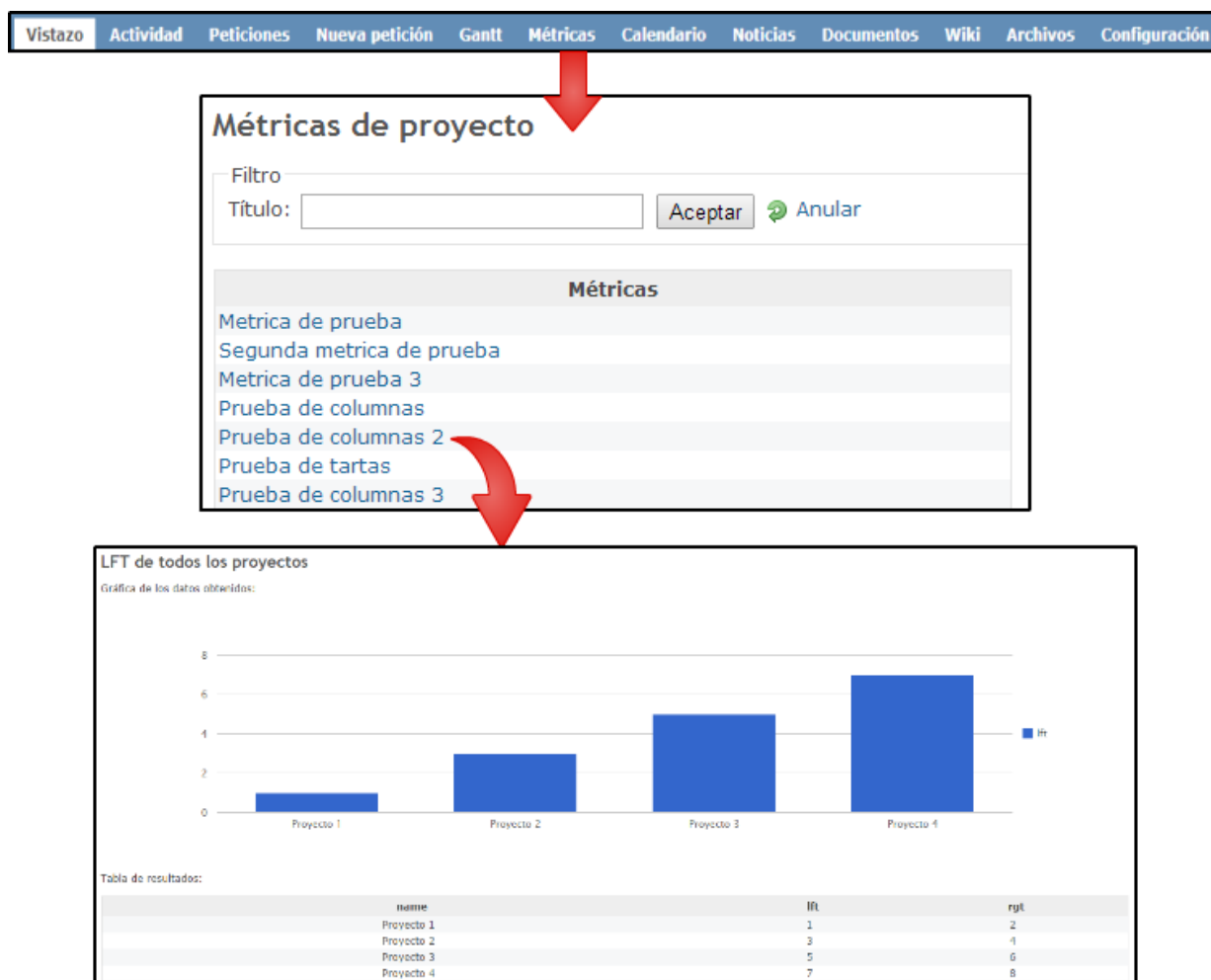


Ilustración 46: Diagrama de navegación de visualización de métricas de proyecto.

5.3.2 Diseño de las Interfaces de Usuario

Hay que señalar que la descripción de las interfaces de usuario ya se realizó en el apartado 4.5 del presente documento. A continuación, se explicarán los principios de diseño empleados en la elaboración de la interfaz de nuestro plug-in:

Heurísticas de Nielsen:

1. **Visibilidad del estado del sistema:** Se informa a los usuarios del estado del sistema, dando una retroalimentación apropiada en un tiempo razonable; como en el caso de creación de una nueva métrica, en el cual se informa al usuario mediante un mensaje que la métrica ha sido creada con éxito.
2. **Utilizar el lenguaje de los usuarios:** Se utiliza el lenguaje de los usuarios, con palabras o frases que le sean conocidas, en lugar de los términos que se utilizan en el sistema, para que al usuario no se le dificulte utilizar el sistema.
3. **Control y libertad para el usuario:** Se proporciona a los usuarios la posibilidad de interactuar libremente con el sistema, por ejemplo, a la hora de crear nuevas métricas, los administradores podrán crear las métricas a su antojo.
4. **Consistencia y estándares:** La estética del plug-in es consistente con la de la aplicación Redmine. Tanto los colores, como los iconos utilizados siguen el estándar marcado por la aplicación.
5. **Prevención de errores:** Se previene la aparición de posibles errores no deseados por el usuario. El caso más destacado de nuestro plug-in se muestra a la hora de querer borrar una métrica. Para asegurarnos de que el administrador quiere eliminar la métrica y que no ha sido un error de este, se procede a realizar una pregunta de confirmación para la eliminación mediante un pop-up.
6. **Minimizar la carga de la memoria del usuario:** Las interfaces de usuario de nuestro plug-in no están cargadas de información, sino que son muy simples para que el usuario pueda reconocer todos los elementos con un simple vistazo.
7. **Flexibilidad y eficiencia de uso:** El sistema permite personalizar acciones frecuentes para garantizar la flexibilidad de uso, como es el caso de la configuración de métricas.
8. **Diálogos estéticos y diseño minimalista:** La interfaz no contiene información que no sea relevante. El diseño de la interfaz del plug-in es minimalista siguiendo el modelo de la aplicación Redmine.
9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** El sistema muestra mensajes de error que ayudan al usuario a reconocer el error y ofrecen información de cómo debe solucionarlo. Este caso lo encontramos a la hora de que un administrador rellene de forma inadecuada el formulario de creación de métricas, en el cual todos los campos son obligatorios.

10. **Ayuda y documentación:** Se ofrecen textos informativos a los usuarios ofreciéndoles ayuda en el manejo del sistema, como es en el caso de las indicaciones previas al rellenar un formulario.

5.4 Diseño de la Capa de Negocio

5.4.1 Arquitectura general de la solución

Como hemos comentado anteriormente, nuestro plug-in sigue una arquitectura software MVC. En Ruby on Rails esto implica que la capa de negocio la conformen los componentes de los controladores.

Hay que destacar que el controlador realiza las operaciones de configuración y de visualización sobre un tipo de métrica. En nuestro caso tenemos dos tipos de métricas (generales y de proyecto), por lo que tendremos un controlador para cada una: “*MetricageneralsController.rb*” para las métricas generales y “*MetricaprojectsController.rb*” para las métricas de proyecto. Además, contamos con el controlador de inicio “*init.rb*”, el cual se encarga de inicializar el plug-in con la configuración contenida en el fichero.

Los controladores contienen métodos de configuración de métricas, que tienen como objetivo gestionar las peticiones procedentes de las vistas, para las acciones de creación, edición y eliminación de métricas de la BBDD. También tienen implementados los métodos encargados de la visualización de métricas que se emplean para enviar a las vistas los datos recogidos de la BBDD.

5.4.2 Diagramas de Clases de Diseño

A continuación se muestra el diagrama UML correspondiente al diagrama de clase de diseño de nuestro plug-in:

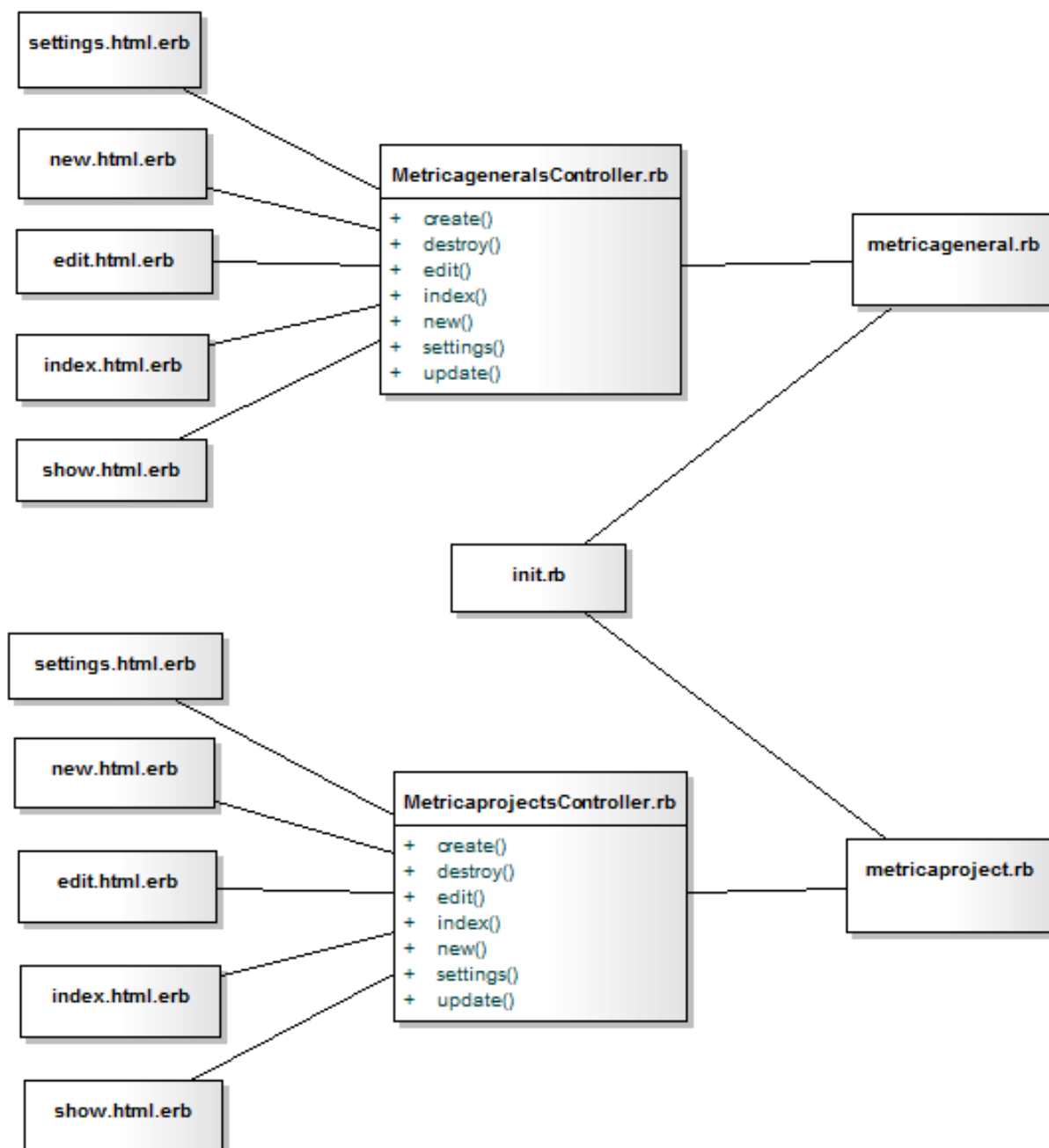


Ilustración 47: Diagrama de clases.

La interacción entre los componentes de nuestro sistema, sigue el marcado por la arquitectura MVC: El usuario interactúa con la interfaz de usuario y el controlador correspondiente recibe

(por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega y accede al modelo, de tal manera que realiza una consulta de la información contenida en la BBDD. El controlador también gestiona las operaciones de acceso al modelo para la creación, modificación o borrado de datos en la BBDD.

Las interfaces de usuario de nuestro Plug-In que corresponden con las vistas de nuestra arquitectura son: *settings*, *new*, *edit*, *index* y *show*, tanto para métricas generales como de proyecto con extensión *“.html.erb”*. Cada una de estas vistas, tiene un método con el mismo nombre de la vista en la clase de su controlador correspondiente, los cuales se encargarán de gestionar las peticiones de dichas vistas.

En el diagrama se puede apreciar la aparición de la clase *“init.rb”*. Esta clase tiene la función de iniciar el Plug-In de métricas en la aplicación Redmine estableciendo la configuración que tenga declarada el fichero. La modificación de esta clase será necesaria para cumplir los requisitos pertenecientes al subsistema de permisos como veremos en el siguiente apartado.

5.4.3 Especificación de clases

En este apartado se detallarán cada una de las clases necesarias para la construcción del plug-in. Como ya comentamos en el apartado 2.1 del presente documento, el sistema seguirá una arquitectura MVC siguiendo la marcada por el framework de Ruby on Rails. Para especificar las clases, las organizaremos en tablas, las cuales recogerán las responsabilidades, los atributos y las operaciones de cada una, así como el componente de la arquitectura al que pertenecen:

metricageneral.rb	
Componente:	Modelo
Responsabilidades:	Contiene los datos de las métricas generales. Además establecerá el nivel de accesibilidad de los datos y sus respectivas validaciones.
Atributos	<p>Como atributos del modelo tenemos las columnas de la tabla métricas:</p> <ul style="list-style-type: none">• título: Nombre de la métrica que vamos a almacenar en la BBDD.• consulta: Consulta en SQL que se va a emplear para obtener los datos de la BBDD.• gráfico: Indica el tipo de gráfico que se va a representar, los cuales pueden ser: columnas, barras, líneas, áreas y tartas.• cantidad: Identifica el tipo de representación que se va a emplear en el gráfico, ya sea simple o multiserie.• serie: Datos que el usuario quiere representar como serie de los gráficos.• categoría: Datos que el usuario quiere representar como etiqueta del eje X de los gráficos.• valor: Conjunto de valores que el usuario desea representar y que supone la etiqueta del eje Y de los gráficos.

metricaproject.rb	
Componente:	Modelo
Responsabilidades:	Contiene los datos de las métricas de proyecto. Además establecerá el nivel de accesibilidad de los datos y sus respectivas validaciones.
Atributos	<p>Como atributos del modelo tenemos las columnas de la tabla métricas:</p> <ul style="list-style-type: none">• título: Nombre de la métrica que vamos a almacenar en la BBDD.• consulta: Consulta en SQL que se va a emplear para obtener los datos de la BBDD.• gráfico: Indica el tipo de gráfico que se va a representar, los cuales pueden ser: columnas, barras, líneas, áreas y tartas.• cantidad: Identifica el tipo de representación que se va a emplear en el gráfico, ya sea simple o multiserie.• serie: Datos que el usuario quiere representar como serie de los gráficos.• categoría: Datos que el usuario quiere representar como etiqueta del eje X de los gráficos.• valor: Conjunto de valores que el usuario desea representar y que supone la etiqueta del eje Y de los gráficos.

MetricageneralsController.rb	
Componente:	Controlador
Responsabilidades:	Es el controlador de métricas generales. Contiene todos los métodos necesarios para gestionar las peticiones de las vistas al modelo “metricageneral” en cuanto a configuración y visualización de métricas.
Operaciones	<ul style="list-style-type: none">• settings: Recoge todas las métricas generales almacenadas en la BBDD para poder mostrar un listado en su vista desde la cual se pueden gestionar todas las métricas generales.• new: Recoge la petición de la vista new, la cual solicita la creación de una nueva métrica.• create: Se encarga de comunicar a la BBDD cuáles son los parámetros que tendrá la nueva métrica para poder crearla. Nos notificará si la operación se ha realizado con éxito y la métrica ha sido creada.• edit: Recoge la petición de editar una métrica identificada por su ID.• update: Se encarga de comunicar a la BBDD los cambios realizados en la métrica y los almacena. Nos notificará si la operación se ha realizado con éxito y la métrica ha sido editada.• destroy: Realiza la operación de eliminar la métrica seleccionada por su ID. Nos notificará si la operación se ha realizado con éxito y la métrica ha sido borrada.• index: Recoge todas las métricas generales almacenadas en la BBDD para poder mostrar un listado en su vista y poder acceder a su visualización.• show: Solicita la recuperación de una métrica en concreto para que su vista pueda mostrar los datos obtenidos.

MetricaprojectsController.rb	
Componente:	Controlador
Responsabilidades:	Es el controlador de métricas de proyecto. Contiene todos los métodos necesarios para gestionar las peticiones de las vistas al modelo "metricaproject" en cuanto a configuración y visualización de métricas.
Operaciones	<ul style="list-style-type: none">• settings: Recoge todas las métricas de proyecto almacenadas en la BBDD para poder mostrar un listado en su vista desde la cual se pueden gestionar todas las métricas de proyecto.• new: Recoge la petición de la vista new, la cual solicita la creación de una nueva métrica.• create: Se encarga de comunicar a la BBDD cuáles son los parámetros que tendrá la nueva métrica para poder crearla. Aquí se comprueba que la consulta SQL contiene el carácter para señalar el "id" del proyecto en el que se encuentra. Nos notificará si la operación se ha realizado con éxito y la métrica ha sido creada.• edit: Recoge la petición de editar una métrica identificada por su ID.• update: Se encarga de comunicar a la BBDD los cambios realizados en la métrica y los almacena. Nos notificará si la operación se ha realizado con éxito y la métrica ha sido editada.• destroy: Realiza la operación de eliminar la métrica seleccionada por su ID. Nos notificará si la operación se ha realizado con éxito y la métrica ha sido borrada.• index: Recoge todas las métricas de proyecto almacenadas en la BBDD para poder mostrar un listado en su vista y poder acceder a su visualización.• show: Solicita la recuperación de una métrica en concreto para que su vista pueda mostrar los datos obtenidos.

A continuación se detallan las clases de las vistas para las métricas generales:

settings.html.erb	
Componente:	Vista
Responsabilidades:	Vista a la que se accede desde la pestaña ‘Configuración de métricas’ del menú superior y seleccionando la opción de métricas generales. Muestra un listado de las métricas generales que existen en la BBDD y las diferentes opciones de configuración que se pueden realizar (crear, editar y borrar). Por otra parte desde esta vista se lleva a cabo la llamada a la operación “destroy” del controlador de métricas generales.

new.html.erb	
Componente:	Vista
Responsabilidades:	Presenta un formulario a rellenar por el usuario para crear una nueva métrica general. El formulario contiene todos los campos para poder crear una métrica general según los atributos que hemos nombrado en la clase del modelo de métricas generales.

edit.html.erb	
Componente:	Vista
Responsabilidades:	Muestra un formulario a rellenar por el usuario para editar la métrica general seleccionada. El formulario contiene todos los campos para poder editar una métrica general según los atributos que hemos nombrado en la clase del modelo.

index.html.erb	
Componente:	Vista
Responsabilidades:	Muestra un listado de todas las métricas generales que están disponibles para su visualización. Cada métrica se muestra con su título correspondiente el cual es un enlace a la vista “show” de esa métrica.

show.html.erb	
Componente:	Vista
Responsabilidades:	Muestra la tabla y el gráfico generados con los datos obtenidos por la métrica general seleccionada.

Para las métricas de proyecto las vistas son prácticamente idénticas como se puede apreciar en las siguientes tablas:

settings.html.erb	
Componente:	Vista
Responsabilidades:	Vista a la que se accede desde la pestaña ‘Configuración de métricas’ del menú superior y seleccionando la opción de métricas de proyecto. Muestra un listado de las métricas de proyecto que existen en la BBDD y las diferentes opciones de configuración que se pueden realizar (crear, editar y borrar). Por otra parte desde esta vista se lleva a cabo la llamada a la operación “destroy” del controlador de métricas de proyecto.

new.html.erb	
Componente:	Vista
Responsabilidades:	Presenta un formulario a rellenar por el usuario para crear una nueva métrica de proyecto. El formulario contiene todos los campos para poder crear una métrica de proyecto según los atributos que hemos nombrado en la clase del modelo de métricas de proyecto.

edit.html.erb	
Componente:	Vista
Responsabilidades:	Muestra un formulario a rellenar por el usuario para editar la métrica de proyecto seleccionada. El formulario contiene todos los campos para poder crear una métrica de proyecto según los atributos que hemos nombrado en la clase del modelo.

index.html.erb	
Componente:	Vista
Responsabilidades:	Muestra un listado de todas las métricas de proyecto que están disponibles para su visualización. Cada métrica se muestra con su título correspondiente el cual es un enlace a la vista “show” de esa métrica.

show.html.erb	
Componente:	Vista
Responsabilidades:	Muestra la tabla y el gráfico generados con los datos obtenidos por la métrica de proyecto seleccionada.

5.5 Diseño de la Capa de Datos

5.5.1 Modelo Físico de Datos

El Plug-In de métricas sólo deberá almacenar los datos de configuración de cada métrica, tanto del tipo de métricas generales, como de métricas de proyecto. Por tanto nuestro modelo de datos del Plug-In se ve compuesto por dos tablas llamadas **metricasGenerals** y **metricasProjects**.

metricasgenerals	metricasprojects
<pre>«column» *PK id :NUMERIC(10,2) * titulo :VARCHAR(80) * consulta :TEXT * grafico :VARCHAR(12) * cantidad :VARCHAR(10) * serie :VARCHAR(15) * categoria :VARCHAR(15) * valor :VARCHAR(15)</pre>	<pre>«column» *PK id :NUMERIC(10,2) * titulo :VARCHAR(80) * consulta :TEXT * grafico :VARCHAR(12) * cantidad :VARCHAR(10) * serie :VARCHAR(15) * categoria :VARCHAR(15) * valor :VARCHAR(15)</pre>
<pre>«PK» + PK_metricaGeneral(NUMERIC)</pre>	<pre>«PK» + PK_metricaProyecto(NUMERIC)</pre>
<pre>«unique» + UQ_metricaGeneral_id(NUMERIC) + UQ_metricaGeneral_titulo(VARCHAR)</pre>	<pre>«unique» + UQ_metricaProyecto_id(NUMERIC) + UQ_metricaProyecto_titulo(VARCHAR)</pre>

Ilustración 48: Modelo físico de datos.

Cabe destacar, que las dos tablas se añaden a la base de datos de forma independiente sin relación con otras tablas. Si se desea consultar el diagrama completo de la base de datos de Redmine, por favor, consulte la siguiente referencia [22].

5.5.2 Descripción de las Entidades de Datos

La capa de datos de nuestro plug-in destaca por estar compuesta de dos tablas que son idénticas. El motivo de esta separación se debe a que de esta forma podemos gestionar mejor las métricas de proyecto, ya que cada proyecto es identificado por su 'id' y necesitaremos modificar la consulta añadiendo ese valor en el momento de ejecutar la consulta que contiene la métrica.

La clave primaria (PK) de nuestras tablas será el ID de la métrica (la cual es generada por defecto por Ruby on Rails). El título de la métrica también es único para cada una por lo que se podría haber elegido como PK, pero sin embargo será nuestra clave alternativa. A continuación se describe cada uno de los atributos de las tablas, para una mejor comprensión:

- **ID:** Este campo se crea automáticamente por el uso del *framework* de Ruby on Rails, e identifica unívocamente cada entrada de la tabla por un número. Este atributo es obligatorio.
- **Título:** Nombre de la métrica. Este atributo es obligatorio y tendrá un límite de 80 caracteres.
- **Consulta:** Cadena de texto que forma la consulta SQL que se empleará para obtener los datos. Este atributo es obligatorio y tendrá un límite de 2000 caracteres.
- **Gráfico:** Almacena el tipo de gráfico que se quiere representar. Este atributo es obligatorio y tendrá un límite de 12 caracteres.
- **Cantidad:** Indica el tipo de representación que se va a utilizar en el gráfico, los cuales pueden ser simple o multiserie. Este atributo es obligatorio y tendrá un límite de 10 caracteres.
- **Serie:** Recoge en forma de cadena de texto la columna de datos que va a suponer la serie a representar en los gráficos. En el caso del gráfico de tartas este campo se omite. Este atributo puede ser nulo, ya que este atributo solo está disponible en gráficos de representación multiserie en los cuales se dispone de más de un valor para una serie. Además, el atributo tendrá un límite de 15 caracteres.
- **Categoría:** Identifica en forma de cadena de texto la columna de datos que va a suponer la etiqueta del eje horizontal de los gráficos. Este atributo es obligatorio y tendrá un límite de 15 caracteres.
- **Valor:** Almacena en forma de cadena de caracteres, la columna de datos que se van a representar y que supone la etiqueta del eje vertical. Este atributo es obligatorio y tendrá un límite de 15 caracteres.

5.6 Diseño de Seguridad

5.6.1 Definición de usuarios, roles y políticas de acceso

Tal y como se indicó en el apartado 3.2 del documento de análisis del proyecto, existen dos tipos de usuario que utilizarán el plug-in de métricas de Redmine. Cada tipo de usuario se diferencia en el nivel de privilegios que tienen para poder visualizar, crear, editar y borrar métricas en Redmine. A continuación se detallan los distintos tipos de usuarios:

- **Usuarios con permisos:** Se tratan de usuarios de Redmine a los que el administrador les ha concedido determinados permisos para poder realizar acciones sobre de las métricas, como pueden ser crear, editar, eliminar y visualizar métricas. Estos permisos fueron detallados en el apartado anterior.

- **Usuarios sin permisos:** Resto de usuarios de Redmine. No tienen privilegios para poder realizar ninguna acción con las métricas. Para que un usuario pueda realizar acciones sobre las métricas el administrador podrá asignarle ciertos permisos.

En el apartado 3.1 del documento de análisis, ya se indicaron los diferentes tipos de permisos que se pueden otorgar a un usuario:

- **Permisos de configuración de métricas:** Estos permisos hacen referencia a la creación, edición y eliminación de métricas en Redmine. Como ya hemos comentado, existen dos tipos de métricas (generales y de proyecto) para las cuales se diferenciarán también sus respectivos permisos, que pueden ser:
 - **Crear métrica:** Este permiso otorga al usuario el poder de crear una nueva métrica.
 - **Editar métrica:** Determina que el usuario puede editar una métrica ya existente en la aplicación Redmine.
 - **Eliminar métrica:** Otorga al usuario el poder de eliminar una métrica.
- **Permisos de visualización de métricas:** También existen dos tipos de permisos relacionados con el tipo de métrica que se desea visualizar:
 - **Ver métricas generales:** El usuario que tenga este permiso podrá visualizar las métricas generales de Redmine.
 - **Ver métricas de proyecto:** El usuario que tenga este permiso podrá visualizar las métricas del proyecto al que están asignados.

5.6.2 Consultas SQL y formularios

Dado que en las métricas se realizan ejecuciones de sentencias SQL, se deberá filtrar adecuadamente dichas sentencias con el objetivo de evitar ataques a la base de datos por parte de personal interno o simplemente por evitar descuidos de los usuarios. Para ello se tendrá que controlar que únicamente se ejecutan sentencias del tipo SELECT, desechando cualquier tipo de sentencia que intervienen en las definiciones de datos, control de acceso a la BBDD y PLSQL. Además, se debe impedir la inserción de código y de caracteres que lo permitan.

Por otro lado tenemos los formularios de creación y edición de métricas, los cuales contienen entradas de texto las cuales deberán estar protegidas frente a ataques XSS.

6 | Plan de pruebas

ID	Subsistema	Categoría	Prueba	Descripción de la prueba	Resultados esperados	Resultado
PU-001	Configuración de métricas	Formulario	Campos obligatorios	Comprobar que el sistema ofrece un mensaje de error cuando no se han rellenado todos los campos obligatorios.	Se muestra un mensaje de error indicando que campos no han sido rellenados o no se han completado correctamente.	OK
PU-002	Configuración de métricas	Formulario	Título de la métrica	Introducir un título de 81 caracteres (máximo permitido 80) y comprobar que muestra un mensaje de error.	Se muestra un mensaje de error indicando que el campo título solo puede tener un máximo de 80 caracteres.	OK
PU-003	Configuración de métricas	Formulario	Consulta SQL de la métrica	Introducir una consulta de 2001 caracteres (máximo permitido 2000) y comprobar que muestra un mensaje de error.	Se muestra un mensaje de error indicando que el campo consulta solo puede tener un máximo de 2000 caracteres.	OK
PU-004	Configuración de métricas	Formulario	Consulta SQL de métrica de proyectos	En edición o creación de métricas de proyecto, Introducir en el campo "consulta SQL" una consulta sin la cadena de texto \$id\$ que identifica el id del proyecto.	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-005	Configuración de métricas	Formulario	Sentencia INSERT	Introducir en el campo "consulta SQL" la cadena de texto INSERT	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK

PU-006	Configuración de métricas	Formulario	Sentencia UPDATE	Introducir en el campo "consulta SQL" la cadena de texto UPDATE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-007	Configuración de métricas	Formulario	Sentencia DELETE	Introducir en el campo "consulta SQL" la cadena de texto DELETE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-008	Configuración de métricas	Formulario	Sentencia CREATE TABLE	Introducir en el campo "consulta SQL" la cadena de texto CREATE TABLE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-009	Configuración de métricas	Formulario	Sentencia DROP TABLE	Introducir en el campo "consulta SQL" la cadena de texto DROP TABLE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-010	Configuración de métricas	Formulario	Sentencia ALTER TABLE	Introducir en el campo "consulta SQL" la cadena de texto ALTER TABLE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-011	Configuración de métricas	Formulario	Sentencia CREATE VIEW	Introducir en el campo "consulta SQL" la cadena de texto CREATE VIEW	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-012	Configuración de métricas	Formulario	Sentencia DROP VIEW	Introducir en el campo "consulta SQL" la cadena de texto DROP VIEW	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-013	Configuración de métricas	Formulario	Sentencia CREATE INDEX	Introducir en el campo "consulta SQL" la cadena de texto CREATE INDEX	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK

PU-014	Configuración de métricas	Formulario	Sentencia DROP INDEX	Introducir en el campo "consulta SQL" la cadena de texto DROP INDEX	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-015	Configuración de métricas	Formulario	Sentencia CREATE	Introducir en el campo "consulta SQL" la cadena de texto CREATE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-016	Configuración de métricas	Formulario	Sentencia DROP	Introducir en el campo "consulta SQL" la cadena de texto DROP	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-017	Configuración de métricas	Formulario	Sentencia SYNONYM	Introducir en el campo "consulta SQL" la cadena de texto SYNONYM	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-018	Configuración de métricas	Formulario	Sentencia GRANT	Introducir en el campo "consulta SQL" la cadena de texto GRANT	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-019	Configuración de métricas	Formulario	Sentencia REVOKE	Introducir en el campo "consulta SQL" la cadena de texto REVOKE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-020	Configuración de métricas	Formulario	Sentencia COMMIT	Introducir en el campo "consulta SQL" la cadena de texto COMMIT	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-021	Configuración de métricas	Formulario	Sentencia ROLLBACK	Introducir en el campo "consulta SQL" la cadena de texto ROLLBACK	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK

PU-022	Configuración de métricas	Formulario	Sentencia DECLARE	Introducir en el campo "consulta SQL" la cadena de texto DECLARE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-023	Configuración de métricas	Formulario	Sentencia OPEN	Introducir en el campo "consulta SQL" la cadena de texto OPEN	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-024	Configuración de métricas	Formulario	Sentencia FETCH	Introducir en el campo "consulta SQL" la cadena de texto FETCH	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-025	Configuración de métricas	Formulario	Sentencia CLOSE	Introducir en el campo "consulta SQL" la cadena de texto CLOSE	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-026	Configuración de métricas	Formulario	Número de consultas	Introducir en el campo "consulta SQL" más de una consulta.	Se muestra un mensaje de error indicando que la consulta SQL no es correcta.	OK
PU-027	Configuración de métricas	Formulario	Serie de la métrica	Introducir un string para el campo "serie" para la gráfica de 16 caracteres (máximo permitido 15) y comprobar que muestra un mensaje de error.	Se muestra un mensaje de error indicando que el campo ejeY solo puede tener un máximo de 15 caracteres.	OK
PU-028	Configuración de métricas	Formulario	Comprobar la serie	Seleccionar una opción en el campo "serie" que no esté en la consulta SQL.	El sistema solo muestra las columnas que se han seleccionado en la consulta.	OK

PU-029	Configuración de métricas	Formulario	Categoría de la métrica	Introducir un string para el campo "categoría" de 16 caracteres o más (máximo permitido 15) y comprobar que muestra un mensaje de error.	Se muestra un mensaje de error indicando que el campo categoría solo puede tener un máximo de 15 caracteres	OK
PU-030	Configuración de métricas	Formulario	Comprobar la categoría	Seleccionar una opción en el campo "categoría" que no esté en la consulta SQL.	El sistema solo muestra las columnas que se han seleccionado en la consulta.	OK
PU-031	Configuración de métricas	Formulario	Valor de la métrica	Introducir un string para el campo "valor" para la gráfica de 16 caracteres o más (máximo permitido 15) y comprobar que muestra un mensaje de error.	Se muestra un mensaje de error indicando que el campo valor solo puede tener un máximo de 15 caracteres.	OK
PU-032	Configuración de métricas	Formulario	Comprobar el valor	Seleccionar una opción en el campo "valor" que no esté en la consulta SQL.	El sistema solo muestra las columnas que se han seleccionado en la consulta.	OK
PU-033	Configuración de métricas	Formulario	Ataque XSS I	Introducir en cualquier campo la cadena de texto SCRIPT.	Se muestra un mensaje de error indicando que el valor introducido en el campo no es válido.	OK
PU-034	Configuración de métricas	Formulario	Ataque XSS II	Introducir en cualquier campo la cadena de texto OBJECT.	Se muestra un mensaje de error indicando que el valor introducido en el campo no es válido.	OK

PU-035	Configuración de métricas	Formulario	Ataque XSS III	Introducir en cualquier campo la cadena de texto APPLETT.	Se muestra un mensaje de error indicando que el valor introducido en el campo no es válido.	OK
PU-036	Configuración de métricas	Formulario	Ataque XSS IV	Introducir en cualquier campo la cadena de texto EMBED.	Se muestra un mensaje de error indicando que el valor introducido en el campo no es válido.	OK
PU-037	Visualización de métricas	Filtros de listas	Listado de métricas generales	El listado de métricas generales se muestra completo.	Se muestran todas las métricas generales.	OK
PU-038	Visualización de métricas	Filtros de listas	Filtro de métricas generales	Comprobar que el filtro de métricas generales funciona correctamente introduciendo un parámetro que forme parte del título de alguna métrica.	Se muestran las métricas generales que contienen en su título el parámetro introducido.	OK
PU-039	Visualización de métricas	Filtros de listas	Filtro de métricas generales	Introducir un parámetro que no forme parte del título de ninguna de las métricas generales.	Se muestra el listado métricas generales vacío.	OK
PU-040	Visualización de métricas	Filtros de listas	Anular filtro de métricas generales	Comprobar que la opción 'Anular' del filtro de métricas generales borra el filtro establecido y muestra el listado completo de métricas generales.	Se elimina el parámetro del filtro y se muestra el listado completo de métricas generales.	OK
PU-041	Visualización de	Filtros de listas	Listado de métricas de proyecto	El listado de métricas de	Se muestran todas las métricas	OK

	métricas			proyecto se muestra completo.	de proyecto.	
PU-042	Visualización de métricas	Filtros de listas	Filtro de métricas de proyecto	Comprobar que el filtro de métricas de proyecto funciona correctamente introduciendo un parámetro que forme parte del título de alguna métrica.	Se muestran las métricas de proyecto que contienen en su título el parámetro introducido.	OK
PU-043	Visualización de métricas	Filtros de listas	Filtro de métricas de proyecto	Introducir un parámetro que no forme parte del título de ninguna de las métricas de proyecto.	Se muestra el listado métricas de proyecto vacío.	OK
PU-044	Visualización de métricas	Filtros de listas	Anular filtro de métricas de proyecto	Comprobar que la opción 'Anular' del filtro de métricas de proyecto borra el filtro establecido y muestra el listado completo de métricas de proyecto.	Se elimina el parámetro del filtro y se muestra el listado completo de métricas de proyecto.	OK
PU-045	Visualización de métricas	Visualizar métricas	Tabla de datos	En la visualización de la métrica se muestra la tabla con los datos obtenidos.	Se muestran los datos de la consulta de forma tabular	OK
PU-046	Visualización de métricas	Visualizar métricas	Gráfico de datos	En la visualización de la métrica se muestra la gráfica con los datos obtenidos.	Se muestran los datos de la consulta de forma gráfica	OK
PU-047	Visualización de métricas	Visualizar métricas	Tipo de gráfico	En la visualización de la métrica se muestra la gráfica del tipo que se había seleccionado en	Se muestra el gráfico del tipo que se había seleccionado.	OK

				su creación o última edición.		
PU-048	Visualización de métricas	Visualizar métricas	Consulta SQL errónea	Comprobar que el sistema muestra una pantalla de error al tratar de visualizar una métrica con una consulta SQL errónea.	Se muestra una pantalla de error que indica que la consulta SQL establecida en la métrica es errónea	OK
PU-049	Configuración de métricas	Creación de métricas	Nueva métrica	Al pulsar sobre la opción 'Nueva métrica', el sistema nos dirige al formulario de creación de una nueva métrica.	Se muestra correctamente la vista del formulario para creación de una métrica nueva	OK
PU-050	Configuración de métricas	Creación de métricas	Creación de una métrica	Rellenar el formulario de creación de una métrica y pulsar el botón 'Crear métrica'.	El sistema muestra un mensaje indicando que la métrica se ha creado y esta aparece en el listado	OK
PU-051	Configuración de métricas	Edición de métricas	Editar métrica	Al pulsar sobre la opción 'Editar', el sistema nos dirige al formulario de edición de la métrica.	Se muestra correctamente la vista del formulario para editar una métrica con los campos rellenados tal cual contenía la métrica	OK
PU-052	Configuración de métricas	Edición de métricas	Edición de una métrica	Comprobar que la métrica se ha editado correctamente y el sistema nos muestra un mensaje indicándolo.	El sistema muestra un mensaje indicando que la métrica se ha editado y esta aparece en el listado actualizada	OK
PU-053	Configuración de métricas	Eliminación de métricas	Confirmación de eliminación	Al pulsar sobre la opción 'Borrar', comprobar que el sistema muestra un "alert" de	El sistema muestra un mensaje de alerta solicitando la confirmación del usuario para	OK

				confirmación de eliminación de la métrica al pulsar la opción borrar.	eliminar la métrica	
PU-054	Configuración de métricas	Eliminación de métricas	Borrar métrica	Al confirmar la eliminación de la métrica, esta se borra de la BBDD.	El sistema muestra un mensaje indicando que la métrica se ha eliminado y no aparece en el listado	OK
PU-055	Configuración de métricas	Filtros de listas	Filtro de configuración de métricas generales	Comprobar que el filtro de configuración de métricas generales funciona correctamente.	Se muestran las métricas generales que contienen en su título el parámetro introducido.	OK
PU-056	Configuración de métricas	Filtros de listas	Filtro de configuración de métricas generales	Introducir un parámetro que no forme parte del título de ninguna de las métricas generales.	Se muestra el listado métricas generales vacío.	OK
PU-057	Configuración de métricas	Filtros de listas	Anular filtro de configuración de métricas generales	Comprobar que la opción 'Anular' del filtro de configuración de métricas generales borra el filtro establecido y muestra el listado completo de métricas generales.	Se elimina el parámetro del filtro y se muestra el listado completo de métricas generales.	OK
PU-058	Configuración de métricas	Filtros de listas	Filtro de configuración de métricas de proyecto	Comprobar que el filtro de configuración de métricas de proyecto funciona correctamente.	Se muestran las métricas de proyecto que contienen en su título el parámetro introducido.	OK

PU-059	Configuración de métricas	Filtros de listas	Filtro de configuración de métricas de proyecto	Introducir un parámetro que no forme parte del título de ninguna de las métricas de proyecto.	Se muestra el listado métricas de proyecto vacío.	OK
PU-060	Configuración de métricas	Filtros de listas	Anular filtro de configuración de métricas de proyecto	Comprobar que la opción 'Anular' del filtro de configuración de métricas de proyecto borra el filtro establecido y muestra el listado completo de métricas de proyecto.	Se elimina el parámetro del filtro y se muestra el listado completo de métricas de proyecto.	OK
PU-061	Visualización de métricas	Menús	Menú superior	Comprobar que el menú superior siempre aparece en su lugar correspondiente en cualquier página del plug-in.	El menú superior aparece en su lugar correspondiente	OK
PU-062	Visualización de métricas	Menús	Menú de proyecto	Comprobar que el menú de proyecto aparece en su lugar correspondiente cuando estamos en la página de visualización de métricas de proyecto.	El menú de proyecto aparece en su lugar correspondiente	OK
PU-063	Permisos	Permisos de visualización	Pestaña 'Métricas generales' visible	Comprobar que la pestaña 'Métricas generales' del menú superior, es accesible por un usuario que tiene permisos de visualización de métricas generales.	Se permite el acceso a la pestaña 'Métricas generales'.	OK

PU-064	Permisos	Permisos de visualización	Pestaña 'Métricas generales' invisible	Comprobar que la pestaña 'Métricas generales' del menú superior, no es accesible por un usuario que no tiene permisos de visualización de métricas generales.	No se permite el acceso a la pestaña 'Métricas generales'.	OK
PU-065	Permisos	Permisos de visualización	Pestaña 'Métricas' visible	Comprobar que la pestaña 'Métricas' del menú de proyecto, es visible por un usuario que tiene permisos de visualización de métricas de proyecto.	La pestaña 'Métricas' se muestra correctamente y se permite el acceso	OK
PU-066	Permisos	Permisos de visualización	Pestaña 'Métricas' invisible	Comprobar que la pestaña 'Métricas' del menú de proyecto, no es visible por un usuario que no tiene permisos de visualización de métricas de proyecto.	La pestaña 'Métricas' no se muestra y no se permite el acceso	OK
PU-067	Permisos	Permisos de configuración	Opción 'Configuración de métricas' visible	Comprobar que la opción 'Configuración de métricas' del menú superior, es accesible por un usuario que tiene permisos de configuración de métricas generales o de proyecto.	Se permite el acceso a la pestaña 'Configuración de métricas'.	OK
PU-068	Permisos	Permisos de configuración	Opción 'Configuración de métricas' invisible	Comprobar que la opción 'Configuración de métricas' del menú superior no es accesible por un usuario que no tiene	No se permite el acceso a la pestaña 'Configuración de métricas'.	OK

				permisos de configuración de métricas generales o de proyecto.		
PU-069	Permisos	Permisos de configuración	Configuración de métricas generales' visible	Dentro de 'Configuración de métricas' la opción de métricas generales, es visible por un usuario que tiene permisos de configuración de métricas generales.	La opción de configuración de métricas generales se muestra correctamente y se permite el acceso	OK
PU-070	Permisos	Permisos de configuración	Configuración de métricas generales' invisible	Dentro de 'Configuración de métricas' la opción de métricas generales, no es visible por un usuario que no tiene permisos de configuración de métricas generales.	La opción de configuración de métricas generales no se muestra y no se permite el acceso	OK
PU-071	Permisos	Permisos de configuración	Configuración de métricas de proyecto' visible	Dentro de 'Configuración de métricas' la opción de métricas de proyecto, es visible por un usuario que tiene permisos de configuración de métricas de proyecto.	La opción de configuración de métricas de proyecto se muestra correctamente y se permite el acceso	OK
PU-072	Permisos	Permisos de configuración	Configuración de métricas de proyecto' invisible	Dentro de 'Configuración de métricas' la opción de métricas de proyecto, no es visible por un usuario que no tiene permisos de configuración de métricas de	La opción de configuración de métricas de proyecto no se muestra y no se permite el acceso	OK

				proyecto.		
PU-073	Permisos	Permisos de configuración	Opción 'Nueva métrica' de métricas generales visible	En la configuración de métricas generales, la opción 'Nueva métrica' es accesible por un usuario que tiene permisos de creación de métricas generales.	La opción de 'Nueva métrica' se muestra correctamente y se permite el acceso	OK
PU-074	Permisos	Permisos de configuración	Opción 'Nueva métrica' de métricas generales invisible	En la configuración de métricas generales, la opción 'Nueva métrica' no es accesible por un usuario que no tiene permisos de creación de métricas generales.	La opción de 'Nueva métrica' no se muestra y no se permite el acceso	OK
PU-075	Permisos	Permisos de configuración	Opción 'Editar' de métricas generales visible	En la configuración de métricas generales, la opción 'Editar' es accesible por un usuario que tiene permisos de edición de métricas generales.	La opción de 'Editar' se muestra correctamente y se permite el acceso	OK
PU-076	Permisos	Permisos de configuración	Opción 'Editar' de métricas generales invisible	En la configuración de métricas generales, la opción 'Editar' no es accesible por un usuario que no tiene permisos de edición de métricas generales.	La opción de 'Editar' no se muestra y no se permite el acceso	OK

PU-077	Permisos	Permisos de configuración	Opción 'Borrar' de métricas generales visible	En la configuración de métricas generales, la opción 'Borrar' es accesible por un usuario que tiene permisos de eliminación de métricas generales.	La opción de 'Borrar' se muestra correctamente y se permite el acceso	OK
PU-078	Permisos	Permisos de configuración	Opción 'Borrar' de métricas generales invisible	En la configuración de métricas generales, la opción 'Borrar' no es accesible por un usuario que no tiene permisos de eliminación de métricas generales.	La opción de 'Borrar' no se muestra y no se permite el acceso	OK
PU-079	Permisos	Permisos de configuración	Opción 'Nueva métrica' de métricas de proyecto visible	En la configuración de métricas de proyecto, la opción 'Nueva métrica' es accesible por un usuario que tiene permisos de creación de métricas de proyecto.	La opción de 'Nueva métrica' se muestra correctamente y se permite el acceso	OK
PU-080	Permisos	Permisos de configuración	Opción 'Nueva métrica' de métricas de proyecto invisible	En la configuración de métricas de proyecto, la opción 'Nueva métrica' no es accesible por un usuario que no tiene permisos de creación de métricas de proyecto.	La opción de 'Nueva métrica' no se muestra y no se permite el acceso	OK
PU-081	Permisos	Permisos de configuración	Opción 'Editar' de métricas de proyecto visible	En la configuración de métricas de proyecto, la opción 'Editar' es accesible por un usuario que tiene permisos de edición de	La opción de 'Editar' se muestra correctamente y se permite el acceso	OK

				métricas de proyecto.		
PU-082	Permisos	Permisos de configuración	Opción 'Editar' de métricas de proyecto invisible	En la configuración de métricas de proyecto, la opción 'Editar' no es accesible por un usuario que no tiene permisos de edición de métricas de proyecto.	La opción de 'Editar' no se muestra y no se permite el acceso	OK
PU-083	Permisos	Permisos de configuración	Opción 'Borrar' de métricas de proyecto visible	En la configuración de métricas de proyecto, la opción 'Borrar' es accesible por un usuario que tiene permisos de eliminación de métricas de proyecto.	La opción de 'Borrar' se muestra correctamente y se permite el acceso	OK
PU-084	Permisos	Permisos de configuración	Opción 'Borrar' de métricas de proyecto invisible	En la configuración de métricas de proyecto, la opción 'Borrar' no es accesible por un usuario que no tiene permisos de eliminación de métricas de proyecto.	La opción de 'Borrar' no se muestra y no se permite el acceso	OK
PU-085	Permisos	Asignación de permisos	Asignar permisos a un usuario	Un usuario logueado con permisos de administración otorgará permisos de visualización y configuración de métricas a un usuario.	Los permisos se han establecido y el usuario puede realizar las acciones que se le han asignado.	OK

PU-086	Visualización de métricas	Exportaciones	Exportar a CSV una métrica general	El usuario seleccionará la opción de exportación a CSV dentro de la visualización de una métrica general.	El fichero se exporta correctamente, almacenado todos los datos expuestos en la tabla.	OK
PU-087	Visualización de métricas	Exportaciones	Exportar a CSV una métrica de proyecto	El usuario seleccionará la opción de exportación a CSV dentro de la visualización de una métrica de proyecto.	El fichero se exporta correctamente, almacenado todos los datos expuestos en la tabla.	OK
PU-088	Visualización de métricas	Exportaciones	Exportar a PDF una métrica general	El usuario seleccionará la opción de exportación a PDF dentro de la visualización de una métrica general.	El fichero se exporta correctamente, mostrando la gráfica y la tabla correspondiente.	OK
PU-089	Visualización de métricas	Exportaciones	Exportar a PDF una métrica de proyecto	El usuario seleccionará la opción de exportación a PDF dentro de la visualización de una métrica de proyecto.	El fichero se exporta correctamente, mostrando la gráfica y la tabla correspondiente.	OK
PU-090	General	Idioma	Idioma del plug-in en castellano	Entrar en Redmine con la opción de idioma en castellano y comprobar que los elementos del plug-in aparecen en castellano.	Todo el plug-in se muestra en castellano.	OK
PU-091	General	Idioma	Idioma del plug-in en inglés	Entrar en Redmine con la opción de idioma en inglés y comprobar que los elementos del plug-in aparecen en inglés.	Todo el plug-in se muestra en inglés.	OK

7 | Manual de usuario

7.1 Introducción

Este apartado pretende servir de guía a los usuarios del Plug-In Métricas para Redmine de SATEC, indicando cómo pueden utilizar la herramienta de forma útil para obtener resultados que mejoren la gestión de proyectos.

Las métricas de Redmine constituyen una forma de medir distintos aspectos de los proyectos que se alojan en Redmine. De esta forma se obtendrá información referente a todos los proyectos en forma de **métricas generales**, así como de solo un proyecto en concreto en el caso de **métricas de proyecto**. La información recogida por las métricas será representada de forma gráfica y tabular.

Este manual define un uso adecuado de la herramienta, aunque no establece un plan cerrado ya que el usuario tendrá libertad para gestionar las métricas según desee, ya que aporta suficiente flexibilidad para adaptarse a las distintas necesidades del usuario. Se establecerán recomendaciones para que el usuario pueda cumplir sus objetivos con el uso de la herramienta.

7.2 Permisos del plug-in

7.2.1 Asignación de permisos

Todas las acciones que los usuarios pueden realizar en Redmine las determinan los permisos que tienen asignados. Por tanto, si los usuarios quieren hacer uso del Plug-In de métricas necesitan que el administrador de Redmine les asigne ciertos permisos.

Para el Plug-In de métricas se cuenta con los siguientes permisos, tanto para las métricas generales, como para las métricas de proyecto.

- **Visualizar métricas generales:** Este permiso permite que el usuario pueda visualizar las métricas generales aplicables a todos los proyectos alojados en Redmine.
- **Visualizar métricas de proyecto:** Permite al usuario al que se le ha aplicado este permiso, la posibilidad de visualizar las métricas de un proyecto. En este caso, además de tener asignado este permiso, el usuario deberá estar asignado al respectivo proyecto sobre el que desea visualizar la métrica y el módulo de métricas debe haber sido activado previamente en ese proyecto (ver apartado 2.2).
- **Configurar métricas generales:** Ofrece la posibilidad al usuario de crear nuevas métricas generales, editar las ya existentes y eliminarlas de la base de datos.

- **Configurar métricas de proyecto:** Permite al usuario la posibilidad de crear nuevas métricas de proyecto, editar las ya existentes y eliminarlas de la base de datos.

La asignación de estos permisos se puede realizar al crear un nuevo perfil, o asignarlos a los roles ya creados. A continuación se muestra una imagen que hace referencia a la asignación de estos permisos en la aplicación Redmine:

Perfiles » Informe de permisos

Permisos	Jefe de proyecto ✓	Desarrollador ✓	Informador ✓
✓ Crear proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Modificar proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Cerrar / reabrir el proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Seleccionar módulos del proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar miembros	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar versiones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Crear subproyectos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Visualizar métricas generales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Configurar métricas generales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Configurar métricas de proyecto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 49: Asignación de permisos.

Como se puede apreciar en la parte inferior de la anterior captura, solo se muestran los permisos para la visualización de métricas generales y la configuración de ambos tipos de métricas. Esto se debe a que en la parte superior de la interfaz se muestran los permisos aplicables a todo Redmine, pero si se desciende a través de la interfaz encontramos los permisos pertinentes a todos los módulos que son aplicables a los proyectos.

■ Métricas	Jefe de proyecto	Desarrollador	Informador
✓ Visualizar métricas de proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 50: Permisos del módulo métricas.


7.2.2 Activación del módulo de métricas

Una vez activado el permiso de visualización de métricas de proyecto, se debe activar el módulo en el proyecto sobre el que se quiere aplicar. Para ello, dentro del menú del proyecto en cuestión, se debe seleccionar la opción “Configuración > Módulos” y activar el módulo “Métricas” guardando los cambios.

Configuración

[Información](#) [Módulos](#) [Miembros](#) [Versiones](#) [Categorías de las peticiones](#)

Seleccione los módulos a activar para este proyecto:

- ☒ Peticiones
- ☒ Control de tiempo
- ☒ Noticias
- ☒ Documentos
- ☒ Ficheros
- ☒ Wiki
- ☒ Repositorio
- ☒ Foros
- ☒ Calendario
- ☒ Gantt
- ☒ Métricas 

[Seleccionar todo](#) | [No seleccionar nada](#)

[Guardar](#)

Ilustración 51: Activación de módulos.

7.2.3 Script de activación del módulo

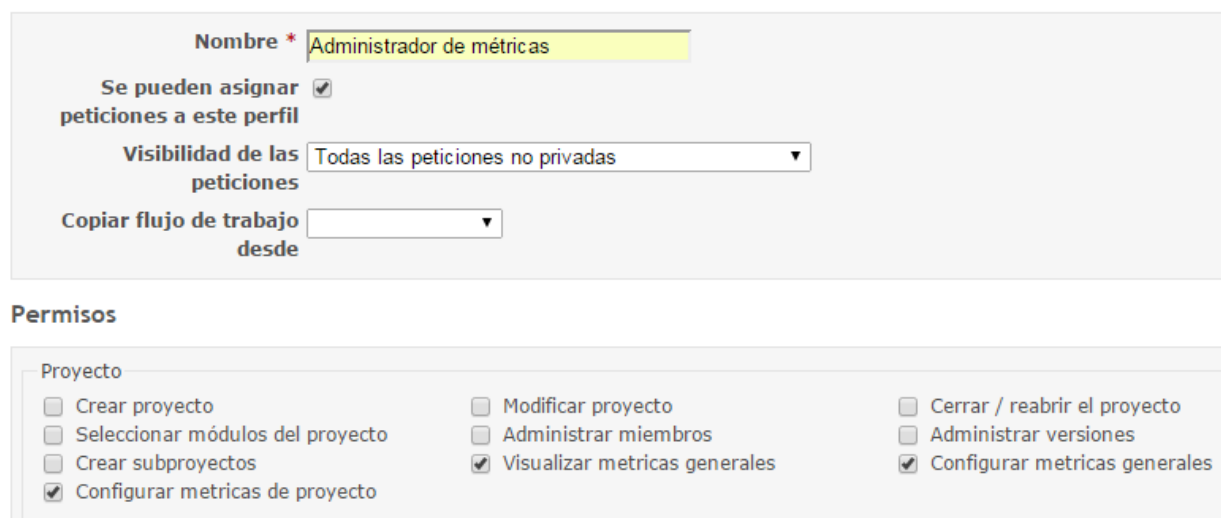
Dentro del plug-in, de forma adicional, se incluye un script de activación del módulo de métricas en todos los proyectos ya existentes en Redmine. Para ejecutar el script **enableModules.rb** se deberá ejecutar el siguiente comando en el directorio raíz donde se encuentra instalado Redmine:

```
ruby script/rails runner -e production enableModules.rb
```

7.2.4 Administrador de métricas

Para una mejor gestión de las métricas, se recomienda crear un perfil de administrador de métricas que contenga los permisos referentes a la configuración de métricas, tanto para las métricas generales, como para las métricas de proyecto. Para ello, el administrador deberá crear un nuevo perfil desde Administración > Perfiles y permisos > Nuevo perfil:

Perfiles » Nuevo perfil



Nombre * Administrador de métricas

Se pueden asignar peticiones a este perfil ☒

Visibilidad de las peticiones Todas las peticiones no privadas ▼

Copiar flujo de trabajo desde ▼

Permisos

Proyecto

<input type="checkbox"/> Crear proyecto	<input type="checkbox"/> Modificar proyecto	<input type="checkbox"/> Cerrar / reabrir el proyecto
<input type="checkbox"/> Seleccionar módulos del proyecto	<input type="checkbox"/> Administrar miembros	<input type="checkbox"/> Administrar versiones
<input type="checkbox"/> Crear subproyectos	<input checked="" type="checkbox"/> Visualizar métricas generales	<input checked="" type="checkbox"/> Configurar métricas generales
<input checked="" type="checkbox"/> Configurar métricas de proyecto		

Ilustración 52: Creación del administrador de métricas.

Como se aprecia en la ilustración anterior, el administrador de métricas tendrá asignados los permisos de configuración de métricas generales y de proyecto, así como de visualización de métricas generales. Además deberá de tener el permiso de visualización de métricas de proyecto correspondiente al módulo de “Métricas”. En la siguiente imagen se podrá ver como quedarían los permisos asignados:

Perfiles » Informe de permisos

Permisos	Jefe de proyecto ✓	Desarrollador ✓	Informador ✓	Administrador de métricas ✓
✓ Crear proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Modificar proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Cerrar / reabrir el proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Seleccionar módulos del proyecto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar miembros	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Administrar versiones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Crear subproyectos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Visualizar métricas generales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
✓ Configurar métricas generales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
✓ Configurar métricas de proyecto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 53: Permisos del administrador de métricas.

El siguiente paso será asignar el perfil de “Administrador de métricas” a un usuario. Hay que señalar, que el administrador de Redmine deberá asignarle a un usuario el perfil en todos los proyectos donde el administrador de métricas deba tener la posibilidad de visualizar las métricas de proyecto, así como la activación del módulo métricas en el proyecto correspondiente (ver apartado 2.2.). Sin embargo, para que usuario que será el administrador de métricas tenga acceso tanto a la configuración de métricas, como a la visualización de métricas generales, bastará únicamente con que se le asigne el perfil de “Administrador de métricas” en un proyecto cualquiera. Esto último, se debe a que los permisos “configurar métricas generales”, “configurar métricas de proyecto” y “visualizar métricas generales” son permisos que se aplican a todo Redmine y en el momento en que un usuario lo tiene asignado en un proyecto puede utilizarlos de forma general.

7.3 Configuración de métricas

En este apartado se detallan las pautas que deberán seguir los usuarios para poder configurar las métricas de forma correcta. Cabe destacar que para acceder al menú de configuración de métricas el usuario debe tener permisos de configuración de métricas generales o de configuración de métricas de proyecto. El acceso al menú de configuración de métricas se encuentra en el menú superior de Redmine:

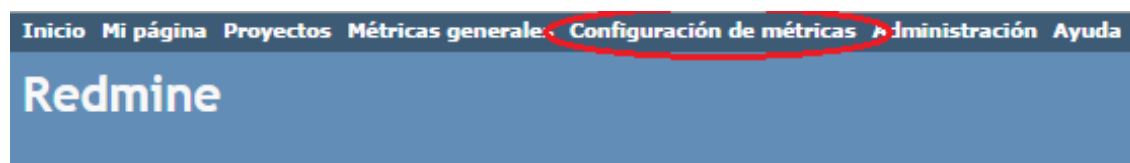


Ilustración 54. Acceso al menú de configuración de métricas

Una vez accedido al menú de configuración de métricas, en él se encuentran las opciones de configurar tanto las métricas generales como las métricas de proyecto.

Configuración de métricas


- Métricas generales
- Métricas de proyecto

Ilustración 55: Menú de configuración de métricas.

Aquí el usuario deberá decidir qué tipo de métricas quiere configurar. En este caso seleccionaremos la opción “Métricas generales”:

Configuración de métricas generales

 Nueva métrica

Filtro
Título:  Anular







Métrica	Opciones
LFT de todos los proyectos	 Editar  Borrar
RGT de todos los proyectos	 Editar  Borrar
Errores en fases de los proyectos	 Editar  Borrar

Ilustración 56: Configuración de métricas generales.

En este menú, el usuario podrá realizar todas las tareas de configuración de métricas generales que se permiten: Creación de nuevas métricas, edición de métricas ya existentes y eliminación de métricas.

7.3.1 Crear nueva métrica

Para crear una nueva métrica, el usuario deberá seleccionar la opción “Nueva métrica” y rellenar el formulario correctamente. Para ello deberá rellenar todos los campos obligatorios (indicados con el símbolo *).

Nueva métrica

(*) Campos obligatorios

Si desea establecer un alias para los nombres de las columnas, utilice las comillas simples. Ejemplo: nombreColumna AS 'nombreAlias'



Formulario de creación de una métrica. El formulario contiene los siguientes campos:

- Título ***: Campo de texto.
- Consulta SQL ***: Área de texto para introducir la consulta SQL.
- Representación ***: Opciones de radio para **Simple** y **Multiserie**.
- Tipo de gráfico ***: Opciones de radio para **Columnas**, **Barras**, **Líneas**, **Área** y **Tarta**.
- Serie ***: Selector desplegable con la opción **Seleccione**.
- Categoría ***: Selector desplegable con la opción **Seleccione**.
- Valor ***: Selector desplegable con la opción **Seleccione**.

Botón: **Crear métrica**

Ilustración 57: Formulario de creación de una métrica.

Hay que destacar que en el campo “Consulta SQL” solo se pueden introducir sentencias tipo SELECT. El resto de sentencias posibles quedan prohibidas por motivos de seguridad. Cabe destacar, que está permitido establecer alias a los nombres de las columnas que se quieren seleccionar para conseguir una representación más legible tanto de las tablas como de los gráficos. La nomenclatura de los alias deberá ser la siguiente: nombreColumna AS nombreAlias (también puede ir el nombre del alias entre comillas simples en el caso de contener más de una palabra).

El tipo de representación que se va escoger tiene que ver con el número de series que se quieren representar. En el caso de representación simple se representa una sola serie, mientras que en representación multiserie se pueden representar dos o más series. A continuación se muestra un ejemplo de representación simple y otro multiserie para gráficos de columnas:

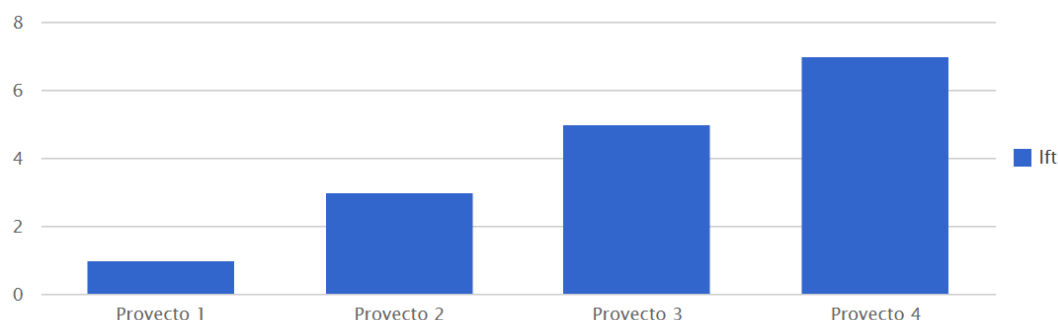


Ilustración 58: Representación simple.

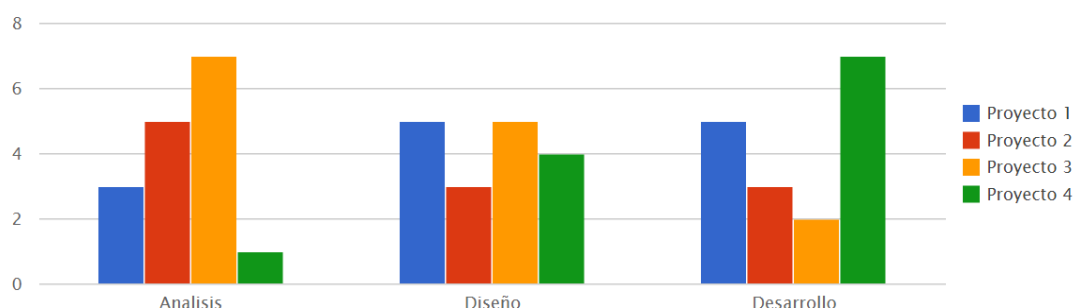


Ilustración 59: Representación multiserie.

Al seleccionar el tipo de representación se producirá una modificación en los campos siguientes que puede rellenar. En el caso de la representación simple, se deshabilita el campo “Serie”, ya que la serie que se va a representar va implícita en el campo “Valor”.

Representación *
☒ Simple
 ☐ Multiserie

Tipo de gráfico *
☐ Columnas
 ☐ Barras
 ☐ Líneas
 ☐ Área
 ☐ Tarta

Serie

Seleccione ▼

Categoría *

Seleccione ▼

Valor *

Seleccione ▼

Ilustración 60: Atributos del gráfico simple.

Tanto los atributos categoría, como valor, se establecen mediante una selección de las columnas (o sus alias) seleccionadas para obtener los datos de la consulta SQL. La categoría establece la etiqueta de los datos numéricos que se van a representar escogidos en el campo

valor. En el caso de la ilustración 10, la categoría son los nombres de los proyectos representados en el eje X del gráfico de columnas.

La representación simple admite todos los tipos de gráficos que se pueden representar (apartado 3.1.1), mientras que para la representación multiserie el gráfico de tipo tarta no se encuentra disponible.

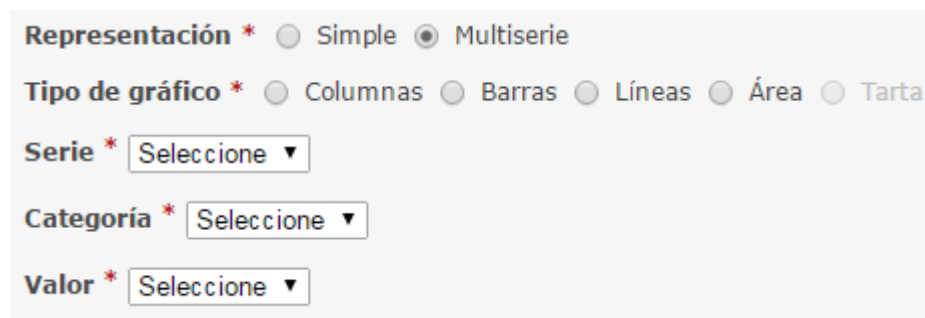


Ilustración 61: Atributos del gráfico multiserie.

En el caso de la representación multiserie, encontramos habilitado el campo “Serie” el cual indica la columna de datos (o su alias) que suponen las series a representar. En el caso de la ilustración 11, la serie son los proyectos sobre los que se aplica, teniendo como categoría las fases de un proyecto (análisis, diseño y desarrollo).

Importante: En el caso de las métricas de proyecto es obligatorio introducir en la consulta SQL los caracteres **\$id\$** para hacer referencia al id del proyecto sobre el que se va a ejecutar la métrica. Por ejemplo: SELECT nombreColumna, FROM projects WHERE id=**\$id\$**.

7.3.1.1 Tipos de gráficos

Para representar los datos de forma gráfica, el plug-in ofrece los siguientes tipos de gráficos:

- **Columnas**

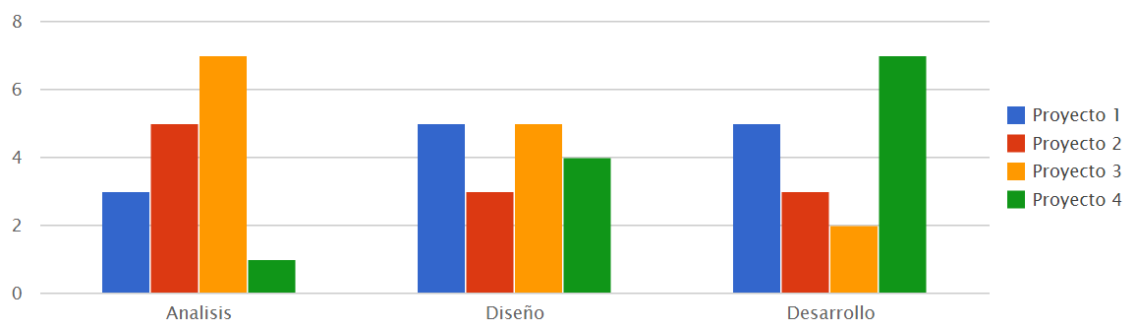


Ilustración 62: Ejemplo de gráfico de columnas.

- **Barras**

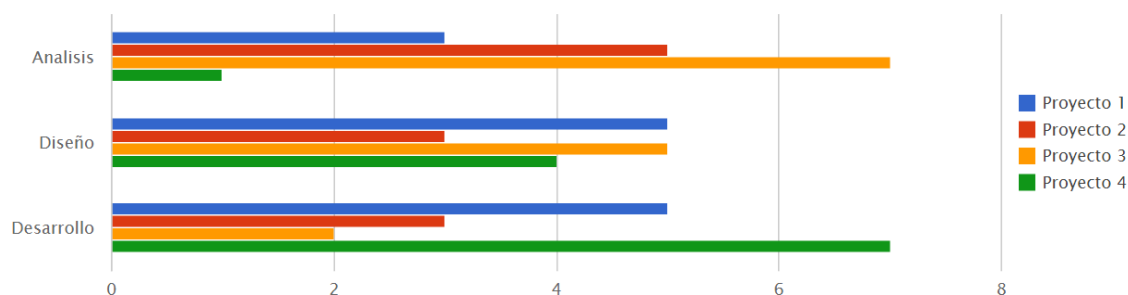


Ilustración 63: Ejemplo de gráfico de barras.

- **Líneas**

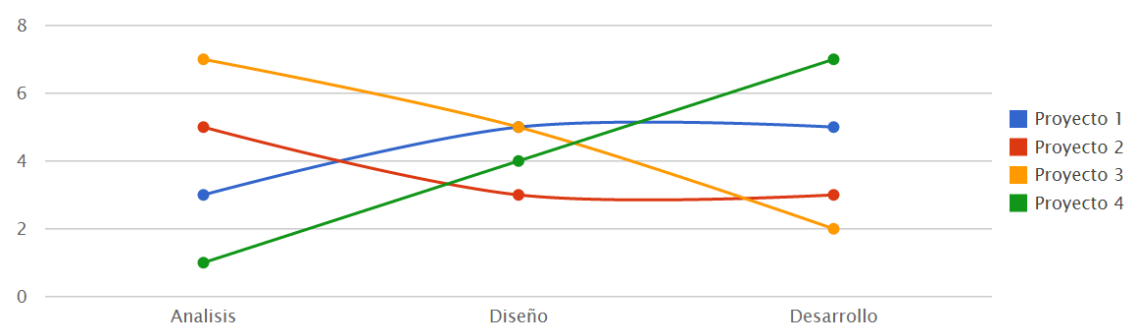


Ilustración 64: Ejemplo de gráfico de líneas.

- **Áreas**

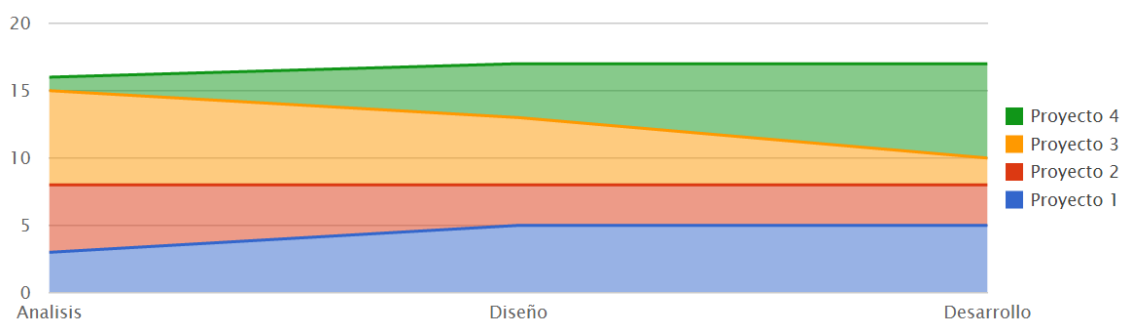


Ilustración 65: Ejemplo de gráfico de áreas.

- **Tartas**

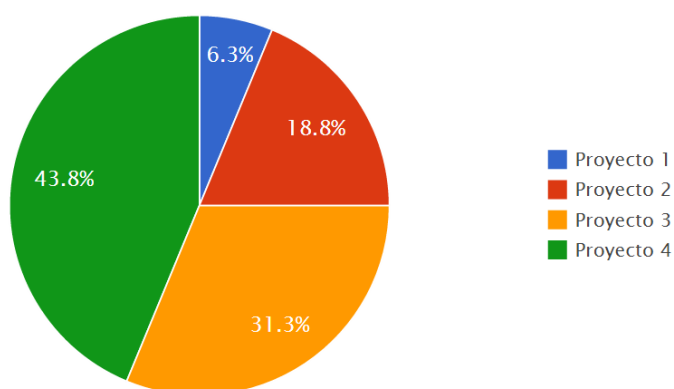


Ilustración 66: Ejemplo de gráfico de tartas.

7.3.2 Editar métrica

Para editar una métrica, el usuario deberá seleccionar la opción “Editar” de la métrica en cuestión que desea modificar. En este caso se accede a un formulario idéntico al de creación de métricas, pero con la particularidad de que los datos de la métrica ya vienen introducidos en sus respectivos campos para su edición.

Editar métrica

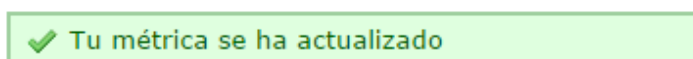
(*) Campos obligatorios

Si desea establecer un alias para los nombres de las columnas, utilice las comillas simples. Ejemplo: nombreColumna AS 'nombreAlias'



Ilustración 67: Formulario de edición de métricas.

Una vez modificados los campos requeridos, el usuario debe pulsar el botón “Actualizar métrica” para que los cambios establecidos en la métrica surtan efecto. Si la métrica ha sido actualizada con éxito, se muestra un mensaje de confirmación:



Configuración de métricas generales

Ilustración 68: Mensaje de confirmación de métrica actualizada.

7.3.3 Eliminar métrica

Para eliminar una métrica de la base de datos, el usuario debe seleccionar la opción “Borrar” de la métrica que desea eliminar. Al pulsar sobre la opción, se mostrará un pop-up pidiendo la confirmación de la eliminación de la métrica.

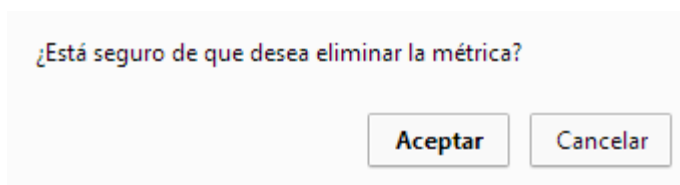


Ilustración 69: Eliminación de métrica.

Si la métrica ha sido eliminada con éxito, se mostrará un mensaje de confirmación en la interfaz de configuración de métricas generales o de proyecto según el tipo:

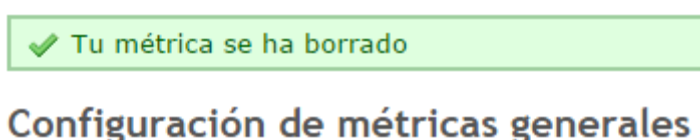


Ilustración 70: Mensaje de confirmación de métrica eliminada.

7.4 Visualización de métricas

En este apartado se detalla el acceso a la visualización de las métricas una vez que ya se hayan creado las métricas generales y de proyecto y se hayan establecido los permisos necesarios para su visualización.

7.4.1 Métricas generales

El acceso al menú de visualización de métricas generales se encuentra en el menú superior de Redmine, seleccionando la opción “Métricas generales”.

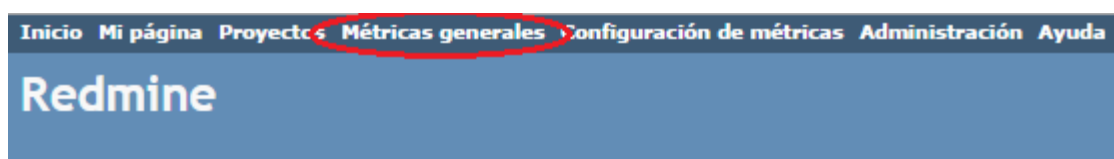


Ilustración 71: Acceso al menú de visualización de métricas generales.

En su interior encontramos un listado de todas las métricas generales que existen en la base de datos.

Métricas generales

Filtro

Título:

Aceptar
Anular

Métricas

LFT de todos los proyectos
RGT de todos los proyectos
Errores en fases de los proyectos

Ilustración 72: Menú de visualización de métricas generales.

Cada título de las métricas es un enlace que lleva a la visualización de esa métrica. Si pulsamos sobre una de ellas podemos ver la representación de los datos obtenidos de forma gráfica (según el gráfico que hayamos escogido al configurarla) y tabular:

LFT de todos los proyectos

Gráfica de los datos obtenidos:

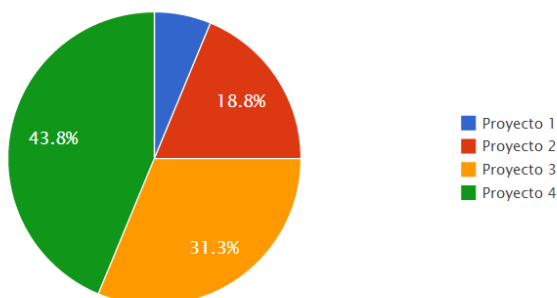


Tabla de resultados:

Nombre	LFT	RGT
Proyecto 1	1	2
Proyecto 2	3	4
Proyecto 3	5	6
Proyecto 4	7	8

Exportar a: [CSV](#) | [PDF](#)

Ilustración 73: Visualización de una métrica.

En esta interfaz, además de la representación de los datos, nos encontramos con las opciones de exportar los datos obtenidos en archivos CSV o PDF (ver apartado 5).

7.4.2 Métricas de proyecto

Para el caso de las métricas de proyecto, el acceso a la visualización de métricas se encuentra dentro del menú de proyecto a través de la pestaña “Métricas”. Cabe destacar que esta pestaña solo será visible y accesible si el usuario tiene permisos de visualización de métricas de proyecto y el módulo “Métricas” ha sido activado en la configuración del proyecto en cuestión.

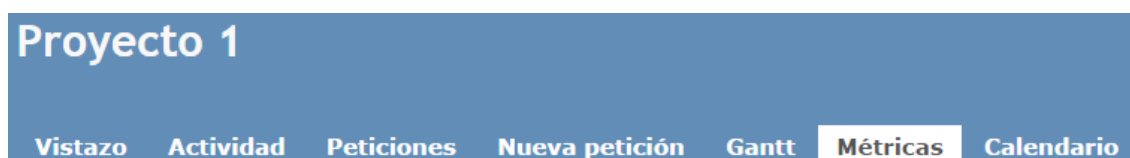


Ilustración 74: Acceso al menú de visualización de métricas de proyecto.

En su interior se encuentra un menú idéntico al de las métricas generales (ilustración 24). Al igual que en el caso de las métricas generales, si seleccionamos la visualización de una métrica accediendo a través del enlace del título de la métrica, se pueden visualizar los datos obtenidos por la métrica (ilustración 25).

7.5 Exportaciones

Para exportar los resultados de una métrica, se deberá acceder a la visualización de una de ellas, ya sea una métrica general o de proyecto (ilustración 25). En la esquina inferior derecha se puede apreciar los distintos formatos de exportación que se ofrecen para la métrica:

Exportar a: CSV | PDF

Ilustración 75: Formatos de exportación.

Hay que destacar que la nomenclatura de los ficheros seguirá el formato siguiente:

- **Métricas generales:** nombreMétrica-fechaArchivo.extensión
- **Métricas de proyecto:** nombreProyecto-nombreMétrica-fechaArchivo.extensión

7.5.1 CSV

Para los archivos con formato CSV se exportarán los datos de forma tabular, tal cual se han representado en Redmine. Además se incluye en el interior el nombre de la métrica, el nombre del proyecto (en el caso de tratarse de una métrica de proyecto) y la fecha y hora de creación del archivo:

	A	B	C
1	Métrica	LFT de todos los proyectos	
2	Fecha	20/10/2014 16:53	
3			
4	name	lft	rgt
5	Proyecto 1	1	2
6	Proyecto 2	3	4
7	Proyecto 3	5	6
8	Proyecto 4	7	8

Ilustración 76: Exportación a CSV.

7.5.2 PDF

En el caso de los archivos con formato PDF se exportarán tanto la representación en forma gráfica como tabular, tal cual se han representado en Redmine. Hay que señalar que el gráfico del tipo áreas no está disponible para la exportación y que en su lugar se exporta en gráfico del tipo líneas. Además se incluye en el interior el nombre la métrica, el nombre del proyecto (en el caso de tratarse de una métrica de proyecto) y la fecha y hora de creación del archivo:

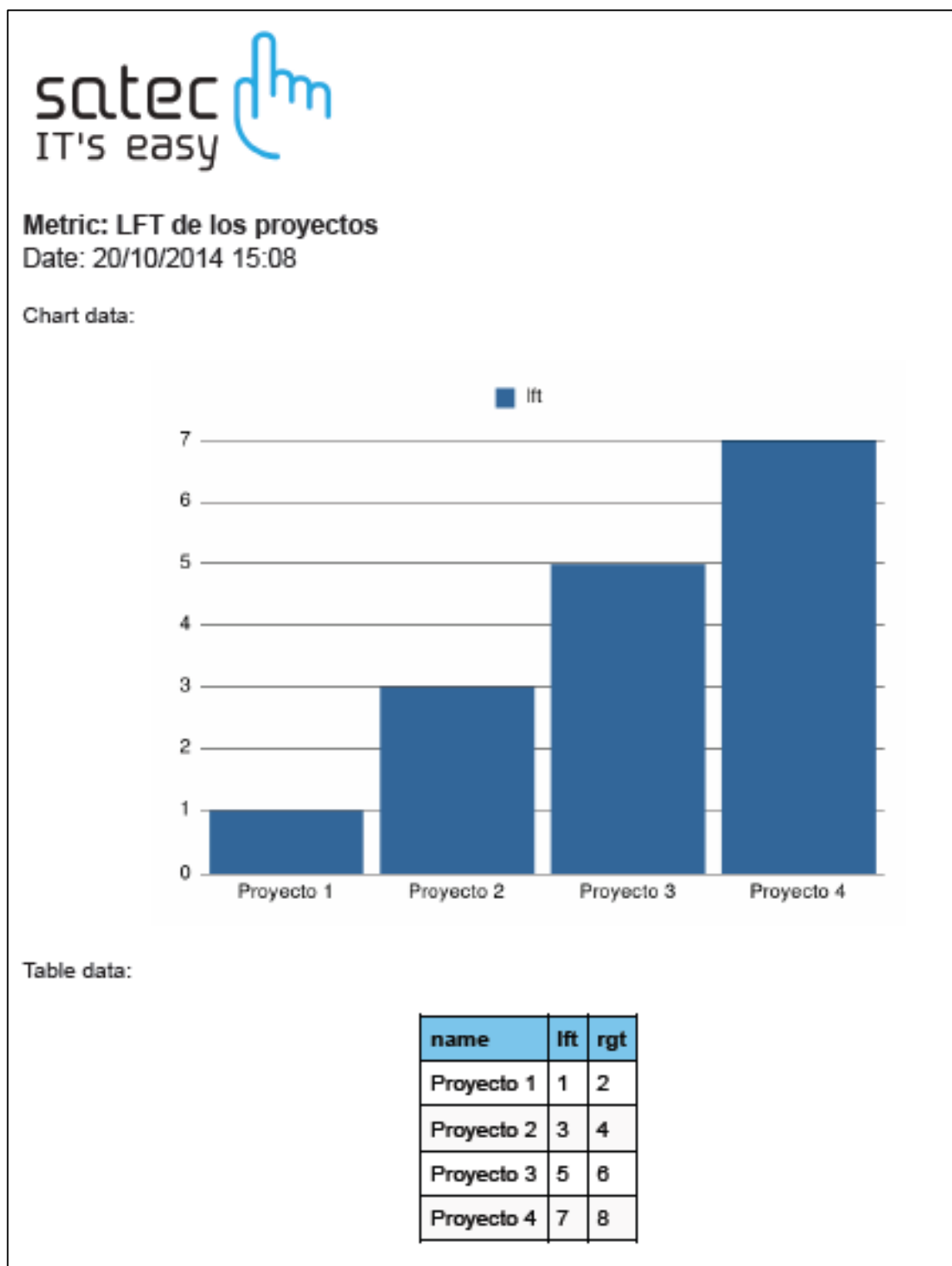


Ilustración 77: Exportación a PDF.

8 | Manual de despliegue

8.1 Introducción

Este apartado pretende servir de guía a la persona encargada de instalar el plug-in en la aplicación Redmine. Se indicarán detalladamente los pasos a seguir para evitar que se produzca algún error durante el proceso de instalación.

Para ello, se deberá preparar el entorno para su instalación mediante la agregación de las gemas de Ruby on Rails necesarias para el correcto funcionamiento del plug-in de métricas. Una vez que se haya preparado Redmine para la instalación del plug-in se procederá a la copia de archivos en su ubicación respectiva, así como la migración de datos necesarios para su funcionamiento.

8.1.1 Consideraciones previas a la instalación

Se recomienda realizar una copia de seguridad de la base de datos de Redmine, así como del directorio donde se encuentre instalado para evitar la pérdida de información en el caso de que surgiese algún error durante la instalación del plug-in.

8.1.2 Directorios de la aplicación

Para facilitar la ubicación de los contenidos de la aplicación y evitar problemas, se establece una nomenclatura común del directorio principal de la aplicación Redmine, ya que dependiendo de la versión de Redmine este directorio tendrá un nombre diferente. Por tanto, para hacer referencia al directorio raíz de la aplicación Redmine utilizaremos el nombre **\$redmine**.

Para hacer referencia al directorio que contiene descomprimido el contenido del plug-in **metricas_redmine.tar.gz**, le llamaremos **\$plugin_home** el cual contiene los directorios `$plugin_home/metricas` y `$plugin_home/pdfs`, así como el archivo `$plugin_home/enableModules.rb`.

8.2 Usuarios

A continuación se definen los diferentes usuarios que intervienen en la instalación del plug-in:

- **\$redmine_installer**: Usuario que instaló Redmine o que posee los mismos permisos.
- **\$web_administrator**: Administrador del servidor web en el que se aloja la aplicación Redmine.

8.3 Instalación de gemas

Lo primero de que se deberá realizar antes de instalar el plug-in, será preparar el entorno de Redmine. Para ello dividiremos esta preparación previa en dos pasos: Agregación de gemas a la aplicación Redmine e instalación de las gemas.

8.3.1 Agregación de gemas

Para agregar las gemas necesarias para el funcionamiento del plug-in, se deberán agregar las siguientes líneas en el archivo `$redmine/gemfile`:

```
gem "chartkick"  
  
gem "prawn"  
  
gem "prawn-table"  
  
gem "gruff"
```

A continuación se detalla para que se emplea cada una de las gemas:

- **Chartkick**: Provee las librerías necesarias para la representación de gráficos en Ruby que serán mostrados en el lado del cliente.
- **Prawn**: Suministra las herramientas necesarias para la exportación de documentos en formato PDF.
- **Prawn-table**: Establece la posibilidad de generar tablas en documentos PDF.
- **Gruff**: Añade las librerías necesarias para poder crear gráficos en Ruby que serán almacenados en el servidor para poder añadirlos a los documentos PDF.

8.3.2 Instalación de gemas

Una vez que se han añadido las gemas se procederá a su instalación. Para ello, se deberá ejecutar el siguiente comando en el directorio `$redmine`, con el usuario `$redmine_installer`. Con este comando, se ha procedido a la instalación del **bundler** de Redmine, el cual ofrece un entorno coherente para proyectos Ruby mediante el seguimiento y la instalación de las gemas y las versiones exactas que se necesitan.

```
$> su - $redmine_installer  
  
$> cd $redmine  
  
$> bundle install --without development test
```

Se instalarán todas las dependencias necesarias y se mostrará un mensaje de confirmación de la instalación:

```
Using rmagick 2.13.2
Using gruff 0.5.1
Using rack-ssl 1.3.4
Using json 1.8.1
Using rdoc 3.12.2
Using thor 0.19.1
Using railties 3.2.19
Using jquery-rails 2.0.3
Using mysql2 0.3.14
Using net-ldap 0.3.1
Using pdf-core 0.4.0
Using ttfunk 1.4.0
Using prawn 1.3.0
Using prawn-table 0.2.0
Using ruby-openid 2.3.0
Using rack-openid 1.4.1
Using bundler 1.7.3
Using rails 3.2.19
Using redcarpet 2.3.0
Your bundle is complete!
Gems in the groups development and test were not installed.
Use `bundle show [gemname]` to see where a bundled gem is installed.
```

Ilustración 78: Instalación de gemas.

8.4 Instalación del plug-in

A continuación se procederá a la instalación del plug-in, una vez que se hayan instalado todas las dependencias necesarias. Hay que destacar, que la instalación la deberá realizar el usuario \$redmine_installer (o que posea sus mismos permisos).

8.4.1 Archivos del plug-in

En este punto de la instalación, se copiarán todos los archivos que componen el plug-in de métricas para Redmine. El paso de estos archivos se realizará en dos pasos. Copia del directorio de “métricas” y del directorio “pdfs”.

8.4.1.1 Directorio métricas

Este directorio contiene el núcleo del plug-in de métricas. Para su instalación correcta, se deberá copiar el directorio “métricas” dentro del directorio \$redmine/plugins.

```
$> su - $redmine_installer

$> cp $plugin_home/metricas $redmine/plugin
```

8.4.1.2 Directorio pdfs

Este directorio contiene las clases encargadas de la generación de documentos en formato PDF en los cuales se exportarán los datos de las métricas. El directorio “pdfs” deberá copiarse en el directorio \$redmine/app.


```
$> su - $redmine_installer  
  
$> cp $plugin_home/pdfs $redmine/app
```

8.4.1.3 Script de activación del módulo

En el plug-in, se incluye de forma adicional, un script de activación del módulo de métricas en todos los proyectos ya existentes en Redmine. Este script es el archivo **enableModules.rb**, el cual deberá copiarse en el directorio `$redmine`. Para ejecutar el script se deberán ejecutar los siguientes comandos:

```
$> su - $redmine_installer  
  
$> cd $redmine  
  
$> ruby script/rails runner -e production  
enableModules.rb
```

8.4.2 Migración de datos

Por último se deberá realizar la migración de la estructura de datos del plug-in para que se generen las tablas necesarias en la base de datos de Redmine. Para ello prepararemos el entorno, estableciendo la variable de entorno en producción con el siguiente comando en el directorio `$redmine`:

```
$> su - $redmine_installer  
  
$> cd $redmine  
  
$> export RAILS_ENV="production"
```

Para finalizar la instalación, se deberá ejecutar el siguiente comando en el directorio `$redmine` y comprobar que las tablas se han creado en la base de datos:

```
$> su - $redmine_installer  
  
$> cd $redmine  
  
$ rake redmine:plugins:migrate
```

8.5 Comprobación de la instalación

Si se han completado todos los pasos y no se ha producido ningún error, ya tendremos el plug-in de métricas instalado correctamente en nuestra aplicación Redmine.

Para confirmar que funciona, se deberá acceder a la aplicación Redmine y comprobar que aparecen en el menú superior las pestañas de “Métricas generales” y “Configuración de métricas”:

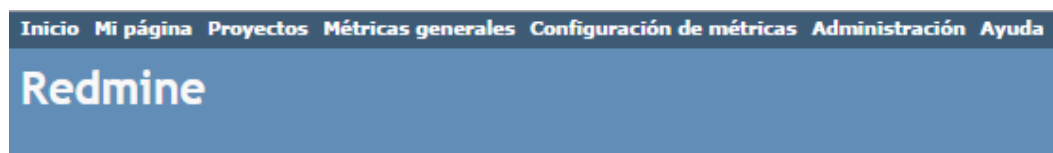


Ilustración 79: Menú superior.

Se recomienda leer el manual de usuario (apartado 7) para conocer como se debe realizar un funcionamiento correcto del plug-in.

8.6 Rollback

En el caso de que surgiese algún error en la instalación o en el funcionamiento del plug-in que afectase a Redmine, será necesario regresar al estado anterior de la instalación. Para ello, se determina el plan de rollback:

1. **Detener el servicio de Redmine:** Antes de realizar cualquier modificación sobre la aplicación de Redmine, es obligatorio que el usuario `$web_administrator` detenga el servicio.

```
$> su - $web_administrator  
  
$> service httpd stop
```

2. **Backup de las tablas *metricasgenerals* y *metricasprojects* (Opcional):** En el caso de que se desee volver a instalar el plug-in y ya se hayan almacenado métricas en la base de datos, se recomienda que el usuario que gestiona la base de datos realice un backup de estas tablas para poder reincorporar los datos en el futuro.
3. **Eliminación de las tablas del plug-in:** Se eliminarán de la base de datos de Redmine las tablas generadas por el plug-in (*metricasgenerals* y *metricasprojects*). El encargado de realizar esta acción será un usuario con permisos de gestión de la base de datos.

```
sql> DROP TABLE metricasgenerals;  
  
sql> DROP TABLE metricasprojects;
```

4. **Eliminación de todos los ficheros del plug-in:** Se eliminará tanto el directorio `$redmine/plugins/metricas`, como el directorio `$redmine/app/pdfs` de Redmine.

Se recomienda que esta acción la realice el mismo usuario que instaló el plug-in.

```
$> su - $redmine_user

$> rm -Rf $redmine/plugins/metricas

$> rm -Rf $redmine/app/pdfs
```

5. **Arrancar el servicio de Redmine:** Por último, el usuario encargado de gestionar el servidor web, deberá iniciar el servicio para que vuelva a estar operativo.

```
$> su - $web_administrator

$> service httpd start
```

Una vez que se han realizado estos pasos el sistema habrá regresado al estado anterior de la instalación del plug-in. Si desea volver a instalar el plug-in deberá volver a ejecutar todos los pasos indicados anteriormente en el documento e importar la copia de seguridad de las tablas *metricasgenerals* y *metricasprojects* en el caso de que haya realizado el paso 1 del plan de rollback.

Hay que destacar que si tras realizar el plan de rollback la aplicación de Redmine no funciona correctamente, será necesario restablecer el sistema con los backups de la aplicación y de la base de datos realizados antes de la instalación. Para ello, se deberán seguir los siguientes pasos:

1. **Detener el servicio de Redmine:** Antes de realizar cualquier modificación sobre la aplicación de Redmine, es obligatorio que el usuario que tenga permisos de gestión del servidor web en el que está alojada la aplicación detenga el servicio.

```
$> su - $web_administrator

$> service httpd stop
```

2. **Restablecer la aplicación:** El usuario que instaló el plug-in, deberá reemplazar el directorio \$redmine al completo por el del backup.
3. **Restablecer la base de datos:** El usuario que gestiona la base de datos deberá reemplazar el contenido de la base de datos de Redmine por el del backup realizado antes de la instalación del plug-in. Para ello se recomienda

eliminar todas las tablas con todo su contenido y ejecutar el script SQL del backup.

4. **Instalación del Bundler:** Se necesita reinstalar el bundler para que actualice las dependencias necesarias para el correcto funcionamiento. Para ello, el usuario encargado de la instalación del plug-in deberá ejecutar el siguiente comando en el directorio \$redmine:

```
$> su - $redmine_user

$> bundle install --without development test
```

5. **Arrancar el servicio de Redmine:** Por último, el usuario que gestiona el servidor web en el que se aloja la aplicación Redmine, deberá iniciar el servicio para que vuelva a estar operativo.

```
$> su - $web_administrator

$> service httpd start
```

9 | Conclusiones

En este apartado se exponen las conclusiones obtenidas durante el desarrollo del proyecto. A continuación, se presentarán resultados obtenidos en la puesta en producción del plug-in, así como mejoras que se podría implantar en una versión posterior e impresiones personales sobre la realización del proyecto.

9.1 Resultados de la puesta en producción

Una vez finalizada la fase de pruebas, me encargaron la labor de instalar el plug-in en una máquina virtual de pre-producción para poder comprobar que el funcionamiento del plug-in era correcto y que no afectaba a los demás componentes que Satec había añadido ya a Redmine. La instalación se realizó con éxito y el sistema pasó todas las pruebas sin que ningún componente de Redmine se viese afectado. Rápidamente el plug-in se puso en producción para que los usuarios pudiesen utilizarlo inmediatamente.

A día de hoy, el plug-in ha sido un éxito a la hora de recopilar información sobre los proyectos, tanto a nivel individual como a nivel colectivo. La empresa ha obtenido resultados muy útiles de los proyectos con la realización de métricas entre las cuales destacan:

- **Errores en cada fase:** Permite obtener información sobre en qué fase del proyecto se producen más errores. Con esta métrica se pretende analizar el origen de los errores para mejorar los procesos y así evitar que se produzcan en fases que pueden resultar críticas posteriormente según avanza el desarrollo del proyecto, como en el caso de la fase de análisis.
- **Uso de recursos:** Recoge información sobre cuantas horas ha dedicado una persona en una fase de un proyecto determinado o en todos los proyectos en los que ha trabajado. De esta forma se puede comprobar si la persona ha realizado un sobreesfuerzo en una fase determinada en la que podría recibir apoyo si se añade la asignación a otra persona.
- **Estado de las peticiones de un proyecto:** Permite comprobar el estado de las peticiones de un proyecto, para obtener información sobre cuántas están sin asignar, cuántas están en curso, cuántas están detenidas y cuántas están finalizadas.
- **Peticiones de un trabajador:** Se trata de una métrica muy similar a la anterior, con la diferencia que esta recoge la información del estado en el que se encuentran las peticiones asignadas a una trabajador.

Los administradores de métricas siguen intentando crear nuevas métricas que les pueda permitir seguir obteniendo información útil. Una de las barreras que les está impidiendo crear nuevas métricas, está siendo la composición de la base de datos de Redmine, la cual recoge información limitada y hace que las consultas sean más complejas cuando se quiere obtener información más profunda de los proyectos.

Además, la incorporación del plug-in ha provocado que la empresa requiera que todos los trabajadores hagan un uso correcto de la herramienta Redmine para poder obtener resultados reales de las métricas, ya que anteriormente los desarrolladores no utilizaban la herramienta o no lo hacían de la forma adecuada porque lo consideraban una pérdida de tiempo. Poco a poco se va concienciando a los trabajadores que hacer un uso correcto de los sistemas de información dedicados a la gestión de proyectos como Redmine ayuda a mejorar los procesos y la productividad de los componentes de la empresa.

9.2 Futuras mejoras

A medida que se iba desarrollando el proyecto, se ocurrían nuevas ideas o mejoras que se podrían aplicar en una segunda versión. Estas mejoras estarían enfocadas a facilitar el proceso de configuración de métricas, así como una exportación de documentos en formato PDF más completa. A continuación, se expone un listado de las posibles mejoras:

- **Información del modelo de datos:** Actualmente el formulario de configuración de métricas permite gestionar la representación de los resultados de la métrica en base a la consulta SQL. La mejora que se pretende añadir tiene como objetivo mejorar la creación de las métricas, de tal manera que durante la realización de la consulta SQL que formará la métrica se produzcan consultas a tiempo real para ofrecer al administrador de métricas las posibles opciones de las que dispone para crear la métrica. Por ejemplo, al introducir el nombre de una tabla en la consulta SQL el sistema mostrará las columnas que componen la tabla, o en el caso de querer realizar una sentencia *NATURAL JOIN* el sistema mostrará el listado de tablas con las que se puede realizar dicha sentencia.
- **Vista previa:** Sería interesante introducir una vista previa de los resultados que se obtienen de la métrica, así como de su representación gráfica y tabular, con el objetivo de agilizar el proceso de configuración de las métricas.
- **Exportación a PDF:** Tiene como objetivo la generación de documentos PDF con una presentación más formal a la actual. Además, se busca generar representaciones de gráficos más precisos que los actuales ya que al usarse una librería diferente a la que genera los gráficos en la web, se han notado diferencias en la forma de representar los datos. Estas diferencias tienen que ver con el espacio reducido que se tiene para su representación en una hoja de formato DIN A4, lo cual hace que no se represente tanta información de los ejes de los gráficos como en la interfaz web, o que los datos se superpongan haciendo confusa la representación y dificultando su visualización.

9.3 Impresiones personales

Personalmente, ha sido gratificante el haber completado el proyecto cumpliendo los plazos marcados por la empresa SATEC. La realización del plug-in “Métricas Redmine” fue un gran reto para mí, al ser el primer proyecto que realizaba en una empresa y al tener que enfrentarme

a tecnologías desconocidas para mí, ya que hasta el momento no había cursado ninguna asignatura en la que se abordasen tecnologías web más allá de HTML5, CSS3 y JavaScript. Por ello, gracias a la realización de este proyecto he adquirido conocimientos de aplicaciones web y servicios web.

Además, hasta la fecha he sido la única persona que ha desarrollado un proyecto en Ruby on Rails, lo cual me hizo ser más autodidacta ya que nadie podía ayudarme a pleno conocimiento. A pesar de ello, no me resultó muy complicada la comprensión de un nuevo lenguaje de programación como Ruby, ya que su sintaxis me resultó muy familiar con Java, el cual ha sido el lenguaje de programación que más he utilizado durante la realización de mi carrera. Por otra parte, las mayores dificultades que he encontrado durante el desarrollo del proyecto ha sido el uso de las librerías para la representación de datos.

10 | Anexo. Abstract en Inglés del Plan de Proyecto.

10.1 Introduction

Currently, information systems have become an essential tool at corporate level. Companies in the software engineering industry need certain information flow to coordinate operational actions, and this information is organized for efficient decision making. In this way the functions and business processes that the company has entrusted to achieve competitive advantage are completed.

What exactly is an information system? An information system can be defined as the set of elements oriented treatment and management of data and information, organized and ready for further use and generated to fulfill an objective. The component parts are:

- Team that develops activities aimed at meeting the target.
- The data to produce the desired information.
- The activities to meet the targets.
- Material resources and information technology.

Nowadays we have to use a large number of tools to help us be more productive with our work, because they allow us to maintain our documentation or tasks ordered. In software engineering in the area of project management tools also allow us to plan projects, track it and measure the degree of fulfillment of the tasks. All this in a single database that organizes information projects for the company to use and achieve the objectives.

Among the tools for project management, you can find free tools and payment tools. The payment tools require the purchase of a temporary license and typically provide as much functionality as free tools.

Companies seek a balance between that offers a payment tool and it offers a free. The companies are studying whether it is worth investing budget to improve the software development process in a collaborative environment. In Satec (Sistemas Avanzados de Tecnología, S.A.) chose a free tool to include sufficient functionality to meet the basic needs of the company for project management. The tool chosen was Redmine.

10.1.1 Redmine



Redmine is an open source software tool developed in **Ruby on Rails** that allows us to manage projects, tasks and also manage and monitor software bugs deliveries from the repository of a software development project [1]. Redmine works with requests whose life cycle is managed by the tool itself. Redmine requests are classified as: functionality, task, error, change, improvement, support, warranty, peer review and auditing CM.

Redmine includes tools to manage projects more visually as a calendar of activities and Gantt charts, which makes it easier to check the timeline of projects.

It highlighted the inclusion of a viewer version control repository (SVN and Git), which collects information on the latest changes in it. It also incorporates the exchange of information between members of the development team, tools like wiki and forum, as well as integration with email.

This tool has become very popular in many companies and is used to manage the planning of a project's tasks, objectives and assigning tasks to members of a team. Redmine has the usual requirements that need a tool like this.

One of the most interesting aspects of Redmine is the ability to install plugins to add new features. Redmine is an open source tool and the developers have published numbers plugins on the official website of Redmine [2]. For example, Redmine Time Recorder is a plug-in that lets you record the time we dedicate to tasks and projects in which we work. Another example is Redmine Notification Extension, an extension for Google Chrome that handles notifications show if any change is produced in the bugs in which we are involved.

There is a list of companies using Redmine tool. These include Open Source software companies like Ruby language and TYPO3 Forge. Also on the list are research companies such as Computer Research Laboratory LIM and the Higher Institute of Aeronautics and Space. There are also public sector companies like French Ministry of Education for EOLE project and the Chilean government for initiation of Open Source code [3].

10.1.2 Ruby language programming



Ruby is a dynamic object-oriented programming language created by Yukihiro Matsumoto and published in 1995. His author blended parts of his favorite languages as Perl and Smalltalk to form a new language that balanced functional programming with imperative programming. He

has often said that he is “trying to make Ruby natural, not simple”. Is simple in appearance, but is very complex inside, just like our human body [4].

Ruby is an open source project, which means that any developer can use, modify or distribute. Therefore, Ruby constantly improved due to the large number of developers using Ruby.

The most important features of Ruby are:

- **Everything is an Object:** In Ruby, everything is an object. Every bit of information and code can be given their own properties and actions. Object-oriented programming calls properties by the name instance variables and actions are known as methods. Ruby's pure object-oriented approach is most commonly demonstrated by a bit of code which applies an action to a number. In many languages, numbers and other primitive types are not objects. Ruby lets you set methods and instance variables to all of its types. This eases one's use of Ruby, since rules applying to objects apply to all of Ruby.
- **Flexibility:** Ruby is a flexible language, since it allows its users to freely alter its parts. Essential parts of Ruby can be removed or redefined, at will. Existing parts can be added upon. Ruby tries not to restrict the coder.
- **Blocks:** Ruby blocks are also seen as a source of great flexibility. A programmer can attach a closure to any method, describing how that method should act. The closure is called a block and has become one of the most popular features for newcomers to Ruby from other imperative languages like PHP or Visual Basic.
- **Mixin:** Ruby is characterized by its simple inheritance. But Ruby knows the concept of modules (called Categories in Objective-C). Modules are collections of methods. Classes can mixin a module and receive all its methods for free.

The power of Ruby, comes in part from its standard library, which defines more than 9,000 methods that can be used in your code. It also includes libraries available to developers called RubyGems which provide many more tools. Quite frequently a more complicated application can be developed in minutes, only through research and the use of an existing gem [5].

Since its public release 1995, in Ruby has drawn devoted coders worldwide. In 2006, Ruby achieved mass acceptance. With active user groups formed in the world's major cities and Ruby-related conferences filled to capacity. Ruby-Talk, the primary mailing list for discussion of the Ruby language, climbed to an average of 200 messages per day in 2006. It has dropped in recent years as the size of the community pushed discussion from one central list into many smaller groups. Ruby is ranked among the top 10 on most of the indices that measure the growth and popularity of programming languages worldwide (such as the TIOBE index). Much of the growth is attributed to the popularity of software written in Ruby, particularly the Ruby on Rails web framework [4].

10.1.3 Framework Ruby on Rails

Ruby on Rails (RoR o Rails) is a web application development framework written in the Ruby language. It is designed to make programming web applications easier by making assumptions about what every developer needs to get started [6]. It allows you to write less code while accomplishing more than many other languages and frameworks. Experienced Rails developers also report that it makes web application development more fun.

When you write a Web application using Rails, almost all of the development you do is in Ruby. You can define your database in Ruby, access your database in Ruby, use Embedded Ruby (ERb) in your templates, and code your business logic in Ruby. This frees your mind to focus on one language to learn and know well [7].

The architecture of the Rails framework is one of its many strengths. It provides a solid foundation upon which you can build your own applications. The Model-View-Controller, or MVC, design pattern, has long been accepted as a better way to architect software applications. MVC is accepted as a better way because it makes applications easier to develop, understand, and maintain. MVC simplifies the implementation of an application by dividing it into several layers, each with a given role and responsibilities. The layers that make up an MVC application are the Model, the View, and the Controller layers. The model layer encapsulates an application's data and business logic for manipulating the data. The controller layer is responsible for figuring out what to do with user and other external input. The view layer provides the user interface of an application and presents the data model [8].

The Rails philosophy includes three major guiding principles:

- **DRY:** Don't Repeat Yourself (DRY) is a philosophy that can be seen throughout the Rails framework. What this philosophy means is that you should not have to repeat yourself in code, configuration, or in many cases even documentation, within a single Web application. For example, in a Rails application, you define your database structure in one place and one place only. You don't have your database structure defined in SQL files, this is done in the model part of the application in Ruby code. This saves you work and prevents errors, as well. When something in your application changes, you only need to make the change in one place. You generally do not need to hunt through a mess of files in a Rails application to make a change to your application.
- **Convention over Configuration:** Rails relies on accepted convention over configuration. A common characteristic of many application frameworks is that you have to configure those using lengthy and complex XML files. Rails does what you might think it should do in most cases without having to specify any configuration. You are also able to override the default behavior of Rails in most areas when you need to do something that may be unconventional. A good example of this philosophy in use within Rails is in the standard Rails routing mechanism. Without having to type a single line of configuration, Rails figures out which classes and methods handle every page request, simply by inspecting the URL. Rails has a standard or conventional format for specifying the URLs your application uses. The name of a controller class, an action method, and a primary key identifying a record being worked with are specified in the URL.

- **Opinionated software:** Rails contains built-in opinions of how Web applications should be developed and designed. Some developers might tell you that an application framework should be completely free of opinions, and designed to be as flexible as possible, accommodating any design decisions that application developers might want to use. Rejecting that view, the developers of Rails have staked out a vision and have taken a definite side on how applications should be developed.

Today, many web applications are built using Ruby on Rails framework detriment of J2EE. The main reasons are:

- Requires **less time and effort** to create a web application and running it.
- The applications developed with Rails require **less configuration**, if you want to apply the configuration the framework incorporates by default.
- Rails requires **less code** to perform the same tasks as J2EE thanks to Ruby.
- **Active Record:** is the model in MVC which is the layer of the system responsible for representing business data and logic. Active Record facilitates the creation and use of business objects whose data requires persistent storage to a database. It is an implementation of the Active Record pattern which itself is a description of an Object Relational Mapping system. Rails makes it easy to perform each of CRUD operations (create, read, update and delete) on your data using built-in Active Record methods. J2EE libraries (such as Hibernate) are often more flexible and more powerful than their counterpart Rails libraries (e.g., Active Record), but at the expense of coding and maintenance overhead.

The creators of Twitter, the microblogging social network based on counts, according to the latest data from October 2014, with 284 million users, chose the Ruby on Rails framework for the development of your web application. Among other applications using Ruby on Rails include GitHub, Assembla, Shopify, Groupon and Bloomberg.

10.1.4 Project objectives

When I arrived at the department of software development Satec (Sistemas Avanzados de Tecnología, S.A.) in July 2014, the company requested me to develop a project that had commissioned them Applications and Services department. The company wants me to develop a plug-in for Redmine. The plug-in should offer users the possibility to obtain project data visually. The project was called "**Metric Redmine**".

In the field of software engineering, a metric is any measure or set of measures to know or estimate the size or other characteristics of software or an information system, usually to make comparisons or for planning of development projects. For Redmine, metrics are performed based on a query data (or data set) in particular on a project, which is interesting to represent graphically to see its evolution. For this, the plug-in should be able to run queries in SQL language on Redmine database (which contains all project data performed in Satec), and

display the results in graphical and tabular form. Of this form, it could generate a set of metrics that should give details of the projects and providing information about their status visually.

Before the project proposal, the Department of Applications and Services provided these metrics into a spreadsheet. In the spreadsheet collected the results of executing queries on the database Redmine (Manager MySQL database system). Once they had the data, they organized the data in a table and graphically represented. This process is added the work of adding the table and graph to a report project results in PDF format. Then the report was delivered to the department of development, and the management of the company. The process was inefficient and they spent much time to obtain the representation of the data.

The project proposal was interesting for the development department, because it allowed the project managers have access to a summary of your project with the results obtained from the metrics. In addition, the ability to generate reports metrics results in PDF format directly from the application was very interesting. They avoid the problems of another process.

The use of this component in a company can allow **CMMI** level 4 is reached, because they are created metric whose results will be used to improve processes. CMMI (Capability Maturity Model Integration) is a model designed to achieve high performance operations. Capability Maturity Model Integration (CMMI) is a process improvement training and appraisal program and service administered and marketed by Carnegie Mellon University and required by many DOD and U.S. Government contracts, especially in software development [9]. The diagram below shows the five levels of the model can acquire an organization:

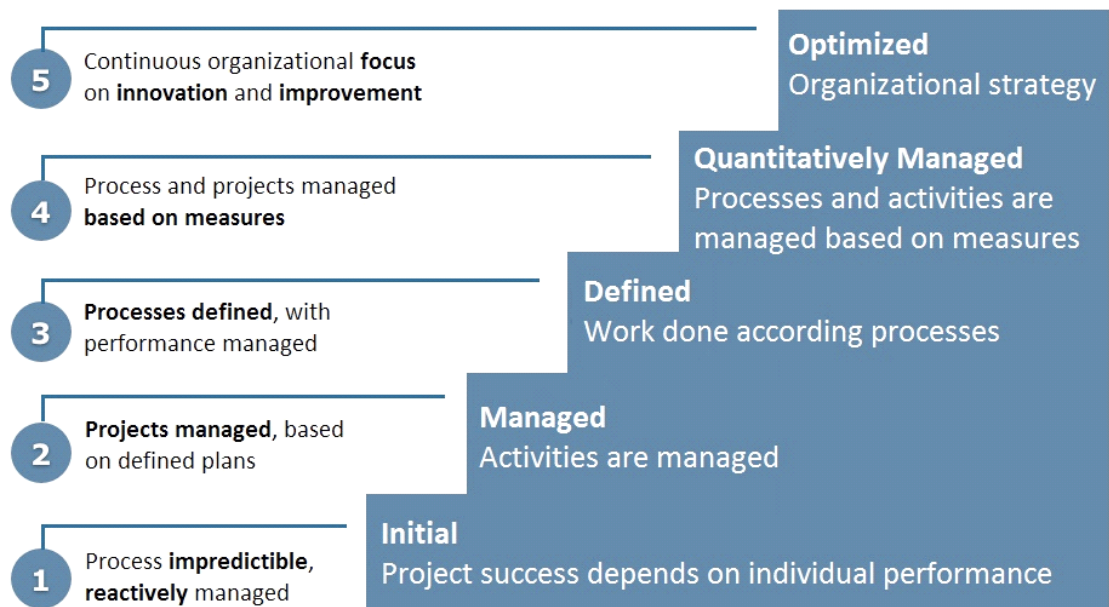


Ilustración 80: CMMI model diagram.

Develop this project was a big challenge for me. For one, I had to learn a new programming language and its respective framework for developing web applications. It would also be my first contact with a project management system within a company.

On the other hand, I had to learn how it works Redmine for later add new functionality. The project was very interesting to me. It was the first project I would develop only. The project would allow me to improve as a professional engineering software development.

Furthermore, no member of the company had developed in Ruby on Rails anything so far. The difficulty of the project was high because I had no support on Rails in the enterprise.

10.1.5 Project Methodology

For the project has followed the methodology of software development in cascade. This model is characterized by following a sequential process phases of the lifecycle of a software project. In this model can't perform the next phase of the project until they have been approved on the last phase [11]. The phases of the software project are:

- **Analysis:** During this phase research is being conducted which includes brainstorming about the software, what it is going to be and what purpose is it going to fulfill.
- **Design:** If the first phase gets successfully completed and a well thought out plan for the software development has been laid then the next step involves formulating the basic design of the software on paper. After the basic design gets approved, then a more elaborated technical design can be planned. Here the functions of each of the part are decided and the engineering units are placed for example modules, programs etc.
- **Implementation:** In this phase the source code of the programs is written.
- **Testing:** At this phase, the whole design and its construction is put under a test to check its functionality. If there are any errors then they will surface at this point of the process.
- **Integration:** in the phase of Integration, the company puts it in use after the system has been successfully tested.
- **Management and Maintenance:** Maintenance and management is needed to ensure that the system will continue to perform as desired.

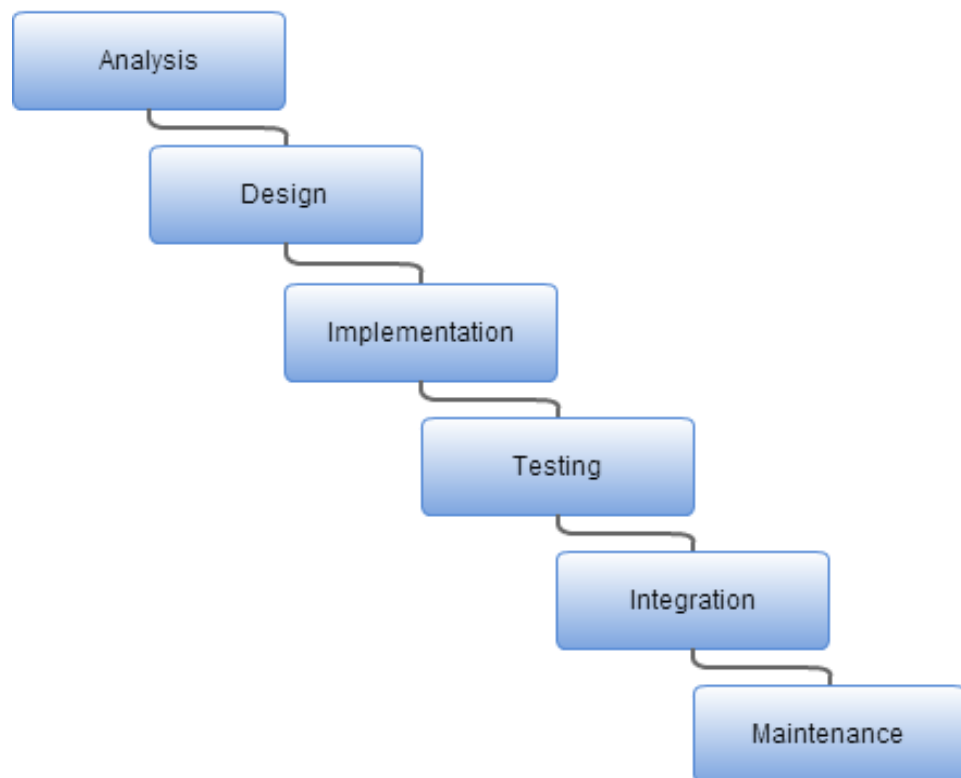


Ilustración 81: Waterfall model diagram.

10.1.6 Document Structure

During the development of project plan document, the sections corresponding to the phases of the software project are added. These sections were performed in SATEC as separate documents, following the rules of the company. In the project plan document have been added to the sections of the feasibility study of the project. In this section the different technologies that can be used to create the plug-in are studied. In addition, the project management section is added to the document and contains project planning and costs. Moreover, user manuals and installation are included.

Sections of the project plan document are:

- **Feasibility study:** In this section the project proposal is studied to analyze its viability. The actual situation of the system is detailed. Other tools competing companies are analyzed and the system to develop is defined. Alternatives that are available for development are studied.
- **Project management:** Project planning with their respective Gantt charts is included. This section indicates the duration of each phase. Furthermore delivery dates indicated. Moreover, the cost of the project is added.

- **Analysis:** The project is analyzed before development, including the specification of requirements, use cases and mockups of user interfaces.
- **Design:** This section describes the system architecture and logic that has occurred during the analysis phase will be defined. System components and their respective explanation will be created. Class diagrams and model of the database are included.
- **Test Plan:** The purpose of this section is to collect all possible tests to run them later once it has completed the development phase. This way you can check the quality of the software and that the system meets all requirements set by the client.
- **User Manual:** This manual helps the user to use the system properly. Includes explanations of the actions the user can perform.
- **Deployment Guide:** This section describes the steps you should follow to install the plug-in correctly in Redmine.

10.2 Conclusions

In this section the conclusions obtained during the project are presented. Next, results of putting into production of plug-in will be presented. Next, results of putting into production of plug-in will be presented. On the other hand, possible improvements that could be implemented in a later version will be explained. Personal impressions of the project are added.

10.2.1 Results in production

Once the testing phase is complete, the company ordered me the task of installing the plug-in in a virtual machine pre-production to verify that the operation of the plug-in was correct and that did not affect the other components that Satec already added to Redmine. The installation was successful and the system passed all tests without any component of Redmine was affected. Quickly the plug-in was put into production so that users could use immediately.

Today, the plug-in has been a success in collecting information on projects, both individually and collectively. The company has obtained useful results of projects performing metrics. The most useful metrics are:

- **Errors in each phase of the project:** Provides information on the project phase where errors occur. This metric allows analyzing the source of errors and can improve project processes. You can avoid that error occur in phases that may be critical later in the project, as in the case of the analysis phase.
- **Use of resources:** Collect information about how many hours a person has spent a phase of a project or all projects that worked. This way you can check if the person has made greater effort in a particular phase in which could be supported if the task is added to another person.

- **State requests of a project:** Check the status of requests for a project. You get Information about how many tasks are unassigned, how many are in progress, many are detained and how many are completed.
- **Requests for a worker:** This is very similar metric to the previous one, except that the state collects information on the state having the worker requests.

Metric administrators are still trying to create new metrics that can allow them to continue to obtain useful information. One barrier that is preventing them from creating new metrics, being the composition of the Redmine database, which collects limited information and makes managers do more complex queries when you want to get deeper information about the projects.

Furthermore, the addition of plug-in has caused the company requires that all workers should make correct use of Redmine tool. If workers do not use the tool correctly, the data will be incorrect. It is necessary that the details are correct to get real results metrics. Previously developers did not use the tool or did incorrectly because they considered it a waste of time. Gradually, workers have been sensitized to make correct use of information systems dedicated to project management helps to improve their work. Processes of development projects and productivity components of the company are also improved.

10.2.2 Future improvements

As the project was developing, I had new ideas or improvements that could be applied in the future in a second version. These improvements would be aimed at facilitating the process of setting metrics, as well as better export documents in PDF format. Following is a list of possible improvements shown:

- **Information data model:** Currently the metric configuration form allows managing the representation of the results of the metric based on the SQL query. The improvement aims to improve creating metrics, such that during the execution of the SQL query that will form the metric, real-time queries to occur to provide the administrator metric of possible options available to create the metric. For example, when we introduce the name of a table in the SQL query the system displays the columns that make up the table. Or in the case of wanting to make a statement NATURAL JOIN the system displays the list of tables you can make that judgment.
- **Preview:** It would be interesting to introduce a preview of the results obtained from the metric, and its graphical and tabular representation, in order to make the setup process metrics faster.
- **Export to PDF:** Aims to generate PDF documents with a more formal presentation than it currently has. In addition, it seeks to generate more accurate representations of the current graphics. We have noticed fails because to the use of different libraries between the web part and PDF export. These differences have to do with the limited space you have for representation on a sheet of A4. Shown incorrectly information axes of the graph or data are overwritten. This makes it difficult the visualization of data.

10.2.3 Personal impressions

Personally, it has been gratifying to have completed the project on schedule set by the company SATEC. The completion of the plug-in "Metric Redmine" was a big challenge for me, being the first project carried out in an enterprise. I've learned to use new technologies which were unknown to me. So far, I only knew the web technologies like HTML5, CSS3 and JavaScript. Therefore, thanks to this project I have acquired knowledge of web applications and web services.

In addition, to date I have been the only person who has developed a project in Ruby on Rails. I have learned to fend for myself. I understood alone how to use a framework because nobody could help within the company. However, I quickly understood a new programming language like Ruby, as its syntax was very familiar with Java, which has been the programming language I most used during the performance of my career.

Moreover, the greatest difficulties I have encountered during the development of the project was the use of libraries for data representation.

11 | Glosario de términos

Término	Descripción
Plug-In	Es un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.
Stakeholders	Personas o entidades con intereses asociados al proyecto.
Framework	Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.
BBDD	Base de datos, en este caso de la aplicación de Redmine
PK	Clave primaria de una tabla de la base de datos.
Mockup	Es un modelo a escala o tamaño real de un diseño, utilizado para la demostración, evaluación del diseño, promoción, y para otros fines.
MVC	Arquitectura software del sistema: Modelo-Vista-Controlador.
PK	Clave primaria de una tabla de la base de datos.
XSS	Cross-site scripting: En un tipo de vulnerabilidad de la seguridad típico de las aplicaciones Web, que permite a una tercera parte inyectar código JavaScript en páginas web vistas por el usuario.

12 | Bibliografía

- [1] Redmine, «Redmine,» [En línea]. Available: <http://www.redmine.org/>. [Último acceso: 30 Enero 2015].
- [2] Redmine, «Redmine Plugins,» [Online]. Available: <http://www.redmine.org/plugins>. [Accessed 30 Enero 2015].
- [3] Redmine, «Who uses Redmine?,» [En línea]. Available: <http://www.redmine.org/projects/redmine/wiki/WeAreUsingRedmine>. [Último acceso: 3 Febrero 2015].
- [4] Ruby Language, «Acerca de Ruby,» [Online]. Available: <https://www.ruby-lang.org/es/about/>. [Accessed 22 Enero 2015].
- [5] L. Ullman, «Why use Ruby?,» de *Ruby: Visual Quickstart Guide*, Peachpit Press, 2008.
- [6] Ruby on Rails, «Getting Started with Rails,» [En línea]. Available: http://guides.rubyonrails.org/getting_started.html. [Último acceso: 15 Enero 2015].
- [7] T. Fisher, «What is Ruby on Rails?,» de *Ruby on Rails Bible*, For Dummies, 2008.
- [8] T. Fisher, «Rails Architecture,» de *Ruby on Rails Bible*, For Dummies, 2008.
- [9] CMMI Institute, «About CMMI Institute,» [Online]. Available: <http://cmminstitute.com/about-cmmi-institute>. [Accessed 1 Febrero 2015].
- [10] Cynertia Consulting, «Auditoría y mejora de procesos TI: CMMi e ITIL,» [Online]. Available: <http://www.cynertiaconsulting.com/consultoria-TIC/auditoria-y-mejora-de-procesos-ti-cmmi>. [Accessed 4 Febrero 2015].
- [11] Waterfall Model, «All About the Waterfall Model,» [Online]. Available: <http://www.waterfall-model.com/>. [Accessed 22 Enero 2015].
- [12] Assembla Inc., «Portfolio Overview,» [Online]. Available: <https://www.assembla.com/portfolio>. [Accessed 1 Febrero 2015].
- [13] Redbooth, «Funcionalidades,» [Online]. Available: <https://redbooth.com/es/funcionalidades/>. [Accessed 1 Febrero 2015].
- [14] O. Smith, «Morris.js,» [Online]. Available: <http://morrisjs.github.io/morris.js/>. [Accessed 6 Febrero 2015].
- [15] Google developers, «Google Charts,» [Online]. Available: <https://developers.google.com/chart/?hl=es>. [Accessed 3 Febrero 2015].
- [16] Highsoft AS, «What is Highcharts?,» [Online]. Available: <http://www.highcharts.com/products/highcharts>. [Accessed 1 Febrero 2015].

- [17] A. Kane, "Chartkick," [Online]. Available: <http://chartkick.com/>. [Accessed 14 Enero 2015].
- [18] G. Grosenbach, "Gruff Graphs," [Online]. Available: <https://github.com/topfunky/gruff>. [Accessed 22 Enero 2015].
- [19] G. Brown, "PrawnPDF," [Online]. Available: <http://prawnpdf.org/api-docs/1.3/>. [Accessed 2 Febrero 2015].
- [20] J. Pace, "PDFKit," [Online]. Available: <https://github.com/pdfkit/pdfkit>. [Accessed 25 Enero 2015].
- [21] M. Z. Sterrett, "Wicked PDF Plugin," [Online]. Available: <http://mileszs.com/wicked-pdf-plugin/>. [Accessed 6 Febrero 2015].
- [22] Redmine, «Database Model,» [En línea]. Available: <http://www.redmine.org/projects/redmine/wiki/DatabaseModel>. [Último acceso: 2 Febrero 2015].